

Data Privacy Management

DPM 2026 – Preproceedings

21st International Workshop on

Data Privacy Management

DPM 2026

Rome, Italy
September 21–25, 2026

Held in conjunction with ESORICS 2026
31st European Symposium on Research in Computer Security

EDITORS

Joaquin Garcia-Alfaro
Guillermo Navarro-Arribas

This volume gathers the preprints of the papers accepted for presentation at DPM 2026, the 21st International Workshop on Data Privacy Management, held in Rome, Italy, on September 21–25, 2026, in conjunction with the 31st European Symposium on Research in Computer Security (ESORICS 2026).

The papers are reproduced here in the form submitted by their authors. They are distributed to the workshop participants for their convenience and do not constitute a formal publication. Copyright of each contribution remains with its authors.

Workshop website: <https://deic.uab.cat/dpm/dpm2026/>

Compiled in August 2026.

Table of Contents

Differential Privacy and Privacy-Preserving Learning

Benchmarking Triangle Counting Algorithms under Local Differential Privacy with PLUTO <i>Long Pham and Stacey Truex</i>	1
Differentially Private Community Detection with Privacy Amplification by Shuffling <i>Xincheng Xu, M.A.P. Chamikara, Thilina Ranbaduge, Thierry Rakotoarivelo, David Smith and Qing Wang</i>	18
Towards Verifiable Continuous-Valued Local Differential Privacy Using Harmless Opening <i>Hiroaki Kikuchi and Takao Murakami</i>	35
Fast Simulation Algorithms for OLH using Binomial Modeling <i>Berkay Kemal Balioglu, Alireza Khodaie and M. Emre Gursoy</i>	52
The Privacy Peak: Non-Monotonic Leakage in Federated Learning <i>Hamza Sajjad, Xavier Lessage, Gabriele Costa and Philippe Massonet</i>	69
Privacy-Preserving Decision Tree Inference under CKKS: A Precision Impact Analysis <i>Madické Diadji Mbodj</i>	86
On Private Knowledge Retrieval and Private Inference for secure multi-agent systems <i>Vicenç Torra and Maria Bras-Amorós</i>	103

Credentials, Anonymity and Secure Protocols

SoK: Data Minimization in Verifiable Credentials (workshop version) <i>Norbert Henseler, Robin Doss, Anh Dinh, Marthie Grobler and Tingmin Wu</i>	120
Regulatorily Aligned Consult-Scoped Anonymous Credentials with Accountable Transparency for EHDS-based Health Data Access <i>Mahdi Akil and Frank Pallas</i>	137
Designing and Verifying a Post-Quantum Protocol for Privacy Critical Applications using ABE and ProVerif <i>Robert-William Evans and Florian Kammüller</i>	154
Kleptographic SETUP against Pseudonym Unlinkability or: check your EDIW Wallet <i>Przemysław Błażkiewicz, Anna Lauks-Dutka, Przemysław Kubiak and Mirosław Kutyłowski</i>	166
On the Applicability of Traffic-splitting Strategies as a Protection Shield against End-to-end Traffic Correlation Attacks in Tor <i>Asya Mitseva, Bozhan Cai and Andriy Panchenko</i>	178

Applied and Regulatory Privacy

From Static Schedules to Consent-Driven Retention: An Approach to GDPR Storage Limitation Enforcement <i>Samuel Opoku Daniels and Haralambos Mouratidis</i>	190
A Privacy-Preserving Architecture for Secure Appliance-Level Demand Response in Virtual Power Plants <i>Joan Ferré-Queralt, Jordi Castellà-Roca, Alexandre Viejo, Nicolás Orduz Álvarez and Takao Tsuji</i>	207
Hidden Figures: Fingerprinting Mobile Phones through Passive Background App Traffic <i>Shiva Azizadeh and Gunes Acar</i>	224
Reverse Engineering for Privacy-Preserving Validation of GWAS Outcomes <i>Yuzhou Jiang, Yuqiao Xu and Erman Ayday</i>	241

Program Committee

Abderrahim Ait Wakrime	UM5R
Ken Barker	University of Calgary
Elisa Bertino	Purdue University
Nora Boulahia-Cuppens	Polytechnique Montréal
Jordi Casas-Roma	Universitat Autònoma de Barcelona
Jordi Castellà-Roca	Universitat Rovira i Virgili
Depeng Chen	Anhui University
Mathieu Cunche	University of Lyon / Inria
Sabrina De Capitani di Vimercati	Universita' degli Studi di Milano
Oscar Esparza	UPC
Sara Foresti	Universita' degli Studi di Milano
Sébastien Gambs	Université du Québec à Montréal
Lorena González Manzano	Universidad Carlos III de Madrid
M. Emre Gursoy	Koç University
Florian Kammüller	Middlesex University London and TU Berlin
Christophe Kiennert	Télécom SudParis
Hiroaki Kikuchi	Meiji University
Seongkwang Kim	Korea University
Evangelos Kranakis	Carleton UniversityComputer Science
Mirosław Kutylowski	NASK-National Research Institute
Romain Laborde	Université de Toulouse
Patrick Lacharme	Ensicaen
Brad Malin	Vanderbilt University
Lukas Malina	Brno University of Technology
David Megías	Universitat Oberta de Catalunya
Chris Mitchell	Royal Holloway, University of London
Javier Parra Arnau	Universitat Politècnica de Catalunya
Isabel Praça	GECAD / ISEP
Cristina Pérez-Solà	Universitat Autònoma de Barcelona
Ruben Rios	University of Málaga
Julián Salas	Universitat Oberta de Catalunya
Pierangela Samarati	Universita' degli Studi di Milano, Italy
David Sanchez	University Rovira i Virgili
Alexandre Viejo	Universitat Rovira i Virgili
Isabel Wagner	University of Basel
Jens Weber	University of Victoria

Lena Wiese

Nicola Zannone

Fraunhofer Institute for Toxicology and
Experimental Medicine
Eindhoven University of Technology

Additional Reviewers

Alarcón, Antonio
Genés-Durán, Rafael
Hernández-Serrano, Juan
Sirant, Kamila
Zaremba, Aleksandra

MORNING SESSION 2

Differential Privacy and Privacy-Preserving Learning

Benchmarking Triangle Counting Algorithms under Local Differential Privacy with PLUTO

Long Pham and Stacey Truex^[0000–0002–8274–645X]

Denison University, Granville OH 43023, USA
{pham_13,truexs}@denison.edu

Abstract. Triangle counting is a fundamental graph analytic used in applications such as community detection, anomaly detection, and network analysis. Recent advances in Local Differential Privacy (LDP) have motivated the development of algorithms for privacy-preserving triangle counting without requiring trusted data collection. However, fair comparison across existing methods remains difficult due to inconsistent evaluation settings, heterogeneous implementations, and limited reproducibility. In this paper, we introduce PLUTO, a unified and extensible benchmarking framework for evaluating locally differentially private triangle counting algorithms under standardized experimental conditions. Using PLUTO, we conduct a systematic empirical evaluation of five state-of-the-art methods across varying privacy budgets, graph sizes, and graph topologies, measuring utility, runtime, and communication overhead. Our findings reveal substantial trade-offs among estimation accuracy, computational cost, and communication efficiency, while demonstrating that algorithm behavior depends strongly on graph structure and experimental configuration. In particular, relative algorithm rankings may vary significantly across graph families, highlighting the importance of diverse and reproducible evaluation settings. We further derive practical insights regarding deployment trade-offs, including the effects of graph densification under randomized response, communication-efficient protocol design, and scalability under varying privacy budgets. Our results highlight the need for standardized benchmarking infrastructure for locally private graph analytics and position PLUTO as a foundation for future research in privacy-preserving graph analysis.

Keywords: Differential Privacy · Triangle Counting · Social Networks

1 Introduction

Counting triangles is a fundamental task in graph analytics used to characterize clustering behavior, network transitivity, community discovery, link prediction, and spam filtering [1]. However, triangle counting often requires collecting sensitive relational data, such as social connections and local neighborhood structure. Privacy-preserving protocols that protect these relations, along with an understanding of their substantial utility, runtime, and communication trade-offs, is therefore critical for graph analytics over user-generated data.

Differential Privacy [3] (DP) has emerged as the standard framework for protecting sensitive information during data collection and analysis through noise injection. In the central model of DP (CDP) a central server injects noise after aggregating participant responses while in the local model (LDP) users perturb their own data before transmission, thereby eliminating the need to trust such a central party. While LDP has been extensively studied with tabular data, graph analytics presents additional challenges because records are structurally dependent. In particular, triangle counting requires reasoning over relationships distributed across multiple participants, making direct adaptation of existing LDP techniques nontrivial.

A growing number of algorithms for triangle counting under LDP has been proposed in recent years. However, fair comparison across methods remains difficult due to inconsistent evaluation settings, heterogeneous implementations, and limited experimental standardization. Existing works often evaluate algorithms using different datasets, privacy budgets, graph configurations, and runtime assumptions and, while publicly available, implementations are inconsistent and span multiple programming environments. For example, in publicly available implementations we observed differences in how noisy node degrees are post-processed, such as whether negative values are truncated to zero prior to estimation. These choices can materially affect empirical results. These challenges motivate a standardized benchmarking framework that supports evaluation under consistent settings while remaining extensible to newly proposed algorithms.

Contributions: Addressing this need, we make the following contributions:

- We present PLUTO ¹ (Privacy-preserving benchmarking for Local and Unbiased Triangle Obfuscation), a unified, extensible benchmarking framework evaluating LDP triangle counting under standardized experimental settings.
- Using PLUTO, we conduct a systematic empirical evaluation of five state-of-the-art LDP triangle counting methods varying privacy budgets, graph sizes, and graph topologies measuring utility, runtime, and communication.
- Our experiments demonstrate that algorithm behavior depends strongly on both graph structure and privacy configuration, revealing significant trade-offs among accuracy, computational cost, and communication efficiency.
- Based on our findings, we derive practical insights regarding the deployment characteristics and trade-offs of existing LDP triangle counting approaches.

2 Preliminaries

2.1 Graphs and Triangle Counting

We denote a graph as $G = (V, E)$, where $V = [n]$ denotes the set of n node indices and $E \subseteq \{(u, v) \mid u, v \in V\}$ the set of undirected edges. For a node $v \in V$, let $N(v)$ denote its neighbor set and let $d_v = |N(v)|$ denote its degree.

Triangle counting is used to characterize local clustering, community structure, and network transitivity. Given three nodes $v_i, v_j, v_k \in V$, a triangle is formed if three corresponding edges (v_i, v_j) , (v_i, v_k) , and (v_j, v_k) are all present

¹ PLUTO will be made publicly available upon acceptance of this paper

in E . The total number of triangles in a graph G is denoted by $\Delta(G)$, while an estimated count of triangles in G is denoted by $\overline{\Delta}(G)$.

Let Δ_v denote the number of triangles incident to node v . Since each triangle is incident to exactly three nodes, the number of triangles in the graph is

$$\Delta(G) = \frac{1}{3} \sum_{v \in V} \Delta_v.$$

Triangle counting becomes particularly challenging under local privacy constraints because no individual participant directly observes an entire triangle. Instead, a participant typically observes at most two of the three edges required to verify triangle existence. Consequently, LDP triangle counting algorithms must infer missing graph structure using noisy reports collected across participants, often introducing substantial utility, runtime, and communication trade-offs.

2.2 Local Differential Privacy

Differential Privacy [3] provides formal guarantees limiting the disclosure of individual information during data collection and statistical analysis. In the central model of DP, users trust a data curator to collect raw information and apply privacy-preserving mechanisms before release. In contrast, the local model of Differential Privacy (LDP) eliminates the need for a trusted curator by requiring each user to perturb their own sensitive data prior to transmission.

Under LDP, the data collector receives only perturbed reports and must recover population-level statistics through statistical estimation. While this model provides stronger protection against disclosure by the data collector, the absence of access to raw data often introduces substantial utility loss, particularly for graph analytics where structural dependencies between users are important.

Definition 1 (ϵ -Local Differential Privacy). *Let $\epsilon \in \mathbb{R}_{\geq 0}$. A randomized algorithm $\mathcal{R}$ satisfies ϵ -LDP if for any two inputs v_1, v_2 in the universe $\mathcal{U}$ and any $y \in \text{Range}(\mathcal{R})$,*

$$\Pr[\mathcal{R}(v_1) = y] \leq \Pr[\mathcal{R}(v_2) = y]e^\epsilon$$

The privacy parameter ϵ , commonly referred to as the privacy budget, controls the privacy-utility trade-off. Smaller values of ϵ provide stronger privacy guarantees by increasing perturbation, often at the cost of reduced accuracy.

Applying LDP to graph analytics requires defining the notion of an individual's private contribution. In graph settings, users commonly represent their local information through adjacency bit vectors or neighbor lists. Following [11], we adopt the notion of ϵ -edge LDP, defined as follows.

Definition 2 (ϵ -edge Local Differential Privacy). *Let $\epsilon \in \mathbb{R}_{\geq 0}$. For any $i \in [n]$, let $\mathcal{R}_i$ with domain $\{0, 1\}^n$ be a randomized algorithm of user v_i . $\mathcal{R}_i$ provides ϵ -edge LDP if for any two neighbor lists $a_i, a_i' \in \{0, 1\}^n$ that differ in one bit and any $S \subseteq \text{Range}(\mathcal{R}_i)$,*

$$\Pr[\mathcal{R}_i(a_i) \in S] \leq \Pr[\mathcal{R}_i(a_i') \in S]e^\epsilon$$

Under ϵ -edge LDP each bit in a participant's adjacency vector is protected individually according to the privacy budget ϵ . Informally, this definition ensures

that an adversary observing a perturbed report cannot confidently infer the presence or absence of any individual edge.

Alternative privacy notions have also been proposed. In particular, decentralized DP (DDP) [15] considers a different privacy assumption regarding edge ownership. Under ϵ -edge LDP, the edge reported by user v_i connecting to v_j and the edge reported by v_j connecting to v_i are treated as distinct secrets, with each participant protecting only their own local observation. In contrast, DDP treats these two views as representing the same secret [10], therefore corresponding to a stronger privacy notion (an algorithm satisfying DDP also satisfies ϵ -edge LDP).

Among the algorithms evaluated in this work, only [15] adopts DDP, while the remaining methods assume edge LDP. However, the stronger privacy guarantees of DDP come with additional assumptions regarding users' knowledge of neighboring relationships, which we revisit in Section 4.6.

2.3 Randomized Mechanisms

Randomized Response [17]

Randomized Response (RR), originally introduced by Warner [17], perturbs binary responses to provide plausible deniability for participants. In graph analytics, RR is widely used to privatize adjacency bit vectors representing users' neighbor lists. Intuitively, RR allows a participant to probabilistically report either the true value of an edge or its complement. This uncertainty protects individual edge existence while enabling unbiased estimation after aggregation.

Following [19], let $B = \{b_1, \dots, b_n\}$ denote an adjacency bit vector and ϵ the privacy budget. The perturbed vector $\tilde{B} = \{\tilde{b}_1, \dots, \tilde{b}_n\}$ is generated as follows:

$$\tilde{b}_i = \begin{cases} b_i & \text{w. p. } \frac{e^\epsilon}{1+e^\epsilon} \\ 1 - b_i & \text{w. p. } \frac{1}{1+e^\epsilon} \end{cases}$$

As ϵ decreases, randomized reports become increasingly noisy, often causing sparse graphs to appear substantially denser after perturbation. As shown in Section 4, this effect can significantly influence runtime and communication cost for triangle counting algorithms operating over noisy graph representations.

Laplace Mechanism

The Laplace mechanism is commonly used to protect real-valued statistics. In LDP triangle counting algorithms, it is frequently applied to quantities such as node degree and incident triangle counts. Given a function f and privacy budget ϵ , the Laplace mechanism outputs

$$f + \text{Lap}\left(\frac{S(f)}{\epsilon}\right)$$

where $S(f)$ is the global sensitivity of the function f .

The amount of injected noise scales with the sensitivity of the query and inversely with the privacy budget. Consequently, reducing sensitivity through clipping or participant selection is a common design strategy in LDP triangle counting algorithms, motivating several of the methods evaluated in Section 4.

3 The PLUTO System

PLUTO is designed to address a key challenge in the evaluation of locally differentially private triangle counting algorithms: the lack of a standardized ex-

perimental pipeline. Existing works evaluate algorithms using different graph datasets, privacy budgets, parameter settings, and implementation assumptions, making direct comparison difficult and often requiring researchers to re-implement prior methods. Small implementation differences, such as negative-value truncation or privacy budget allocation strategies, can substantially affect empirical outcomes while remaining under-reported in published work.

To address these limitations, PLUTO provides a unified and extensible benchmarking framework for evaluating LDP triangle counting algorithms under controlled conditions. PLUTO standardizes dataset pre-processing, parameter specification, metric computation, and runtime/communication accounting while remaining modular enough to support newly proposed algorithms and privacy mechanisms. Figure 1 presents the overall system architecture.

3.1 System Architecture

PLUTO consists of four primary components: (1) data processing, (2) algorithm configuration, (3) parameter specification, and (4) metric evaluation. This separation allows experimental factors to be varied independently, enabling controlled comparisons across algorithms while reducing confounding implementation differences. Figure 1 illustrates the framework architecture.

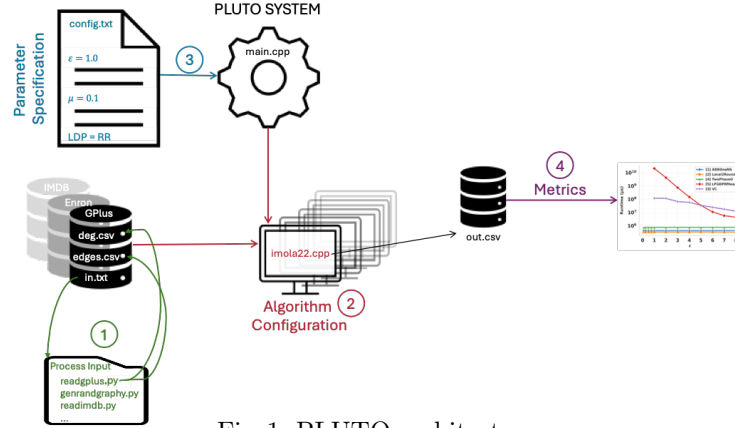


Fig. 1: PLUTO architecture

3.2 Data Processing

Within the data processing component, users may leverage (i) built-in pre-processing support for widely used graph datasets from the Stanford Large Network Dataset Collection (SNAP) [12] (including commonly-used datasets such as Facebook and GPlus [13]), (ii) synthetic graph generation using both Erdős-Rényi [4] and Barabási-Albert [2] models, or (iii) custom graph datasets. This flexibility enables evaluation across both real-world and synthetic graph settings. Unlike works on graph algorithms under Centralized DP models, there is a scarcity of literature in LDP evaluating algorithms on synthetic graphs. Synthetic graph generation enables controlled analysis of how graph topology impacts algorithm behavior under LDP. In particular, varying graph size and edge density allows systematic analysis of utility, runtime, and communication trade-offs under changing structural conditions.

Finally, PLUTO supports custom graph datasets, allowing researchers to evaluate algorithms on domain-specific networks beyond the provided benchmarks. Together, these capabilities enable benchmarking under both realistic and controlled settings while facilitating reproducible experimentation across diverse graph structures.

3.3 Algorithm Configuration

To support newly proposed algorithms, PLUTO adopts a modular design. Each algorithm is implemented as a self-contained module consisting of a driver and any associated subroutines or variants. Modules expose a standardized execution interface for parameter ingestion, metric reporting, and experimental evaluation, ensuring consistent benchmarking across methods.

This modular architecture minimizes implementation overhead when extending the framework. Integrating a new algorithm requires only the addition of a corresponding module and minimal updates to the framework configuration, enabling PLUTO to evolve alongside the rapidly growing literature on locally differentially private graph analytics.

3.4 Parameter Specification

Many locally differentially private triangle counting algorithms additionally require tuning of algorithm-specific parameters. For example, the methods in [10, 11] use an edge clipping constant to limit node degrees before perturbation. Edge clipping removes neighbors from high-degree nodes so that node degree remains below a specified threshold, thereby reducing communication and computation costs while affecting utility. PLUTO supports systematic tuning of such algorithm-specific parameters alongside more general settings, including the privacy budget ϵ , through a centralized and customizable configuration interface.

To further support the full breadth of desired experimental analysis, our parameter specification component is not solely limited to numeric parameters but also allows for users to specify the local differential privacy mechanism. While randomized response [17] remains the dominant approach for achieving local differential privacy in triangle counting, there has been consistent work in the development of alternative local differential privacy mechanisms in recent years [16, 5, 6]. PLUTO currently provides support for both randomized response [17] and optimized unary encoding [16], enabling comparative analysis of how mechanism choice affects triangle counting performance. This modularity further supports extension to newly proposed mechanisms as the literature evolves.

3.5 Metric Evaluation

PLUTO evaluates algorithms along three dimensions: accuracy, runtime, and communication overhead. Together, these metrics characterize utility and computational and practical deployment costs of LDP triangle counting methods.

Accuracy

Accuracy is evaluated using both Relative Error and L2-loss of estimated triangle counts. These metrics capture complementary aspects of utility by measuring

normalized and absolute deviation from the true graph statistic. Prior to algorithm execution, the true number of triangles $\Delta(G)$ is computed for the input graph. Let $\overline{\Delta}(G)$ denote the estimated triangle count returned by an algorithm. We then compute the Relative Error, given by $|\Delta(G) - \overline{\Delta}(G)| / |\Delta(G)|$ while L2-loss is computed as $(\overline{\Delta}(G) - \Delta(G))^2$.

Runtime

Runtime is measured to evaluate computational efficiency and scalability of each method. Since local differential privacy often introduces additional perturbation and aggregation costs, runtime can vary substantially across protocols depending on graph size, privacy budget, and communication structure. PLUTO records end-to-end execution time for each algorithm, enabling systematic comparison of computational requirements under varying experimental settings.

Communication Overhead

Communication efficiency is a critical consideration in locally differentially private protocols, where participants and the data collector frequently exchange perturbed graph information across multiple rounds. In triangle counting, each participant directly observes only two of the three edges involved in a triangle, often requiring additional graph information to estimate missing relationships. Communication of this additional graph information is therefore critical to understanding the usability of a given algorithm. To evaluate practical deployment costs, PLUTO records the total amount of data transferred between participants and the data collector, aggregating both upload and download costs across protocol rounds. Communication overhead is measured in bytes using the data types required by each protocol, including integer-valued graph statistics and real-valued perturbed quantities.

4 Experimental Results and Insights

We evaluate the utility, runtime, and communication overhead of representative locally differentially private triangle counting algorithms using PLUTO. Our goal is not only to compare performance across methods, but also to understand how graph structure, privacy budget, and communication design affect algorithmic behavior under local differential privacy.

4.1 Algorithms

Local2Rounds

Local2Rounds [10] is a two-round protocol that incorporates edge clipping to reduce sensitivity. In the first round, participants perturb and report their neighbor lists using randomized response, allowing the server to construct a noisy graph G' . Since each participant directly observes only two of the three edges in a triangle, users leverage G' to infer the missing edge and estimate the number of incident triangles in the second round. To reduce the sensitivity of the Laplacian noise added to triangle count reports, Local2Rounds randomly removes edges so that node degree does not exceed the maximum degree in G' .

ARROneNS

ARROneNS [11] extends Local2Rounds with a stronger emphasis on communication efficiency. Rather than requiring users to download the entire noisy graph during the second round, ARROneNS employs Asymmetric Randomized Response to sample and perturb edges before transmission. Communication costs are further reduced by selectively downloading only a subset of relevant edges instead of the full graph. Among the variants proposed in [11], we evaluate ARROneNS because it achieves the lowest estimation error. ARROneNS additionally incorporates a double clipping strategy to reduce global sensitivity, although in our experiments we did not observe instances where the clipping threshold was reached.

LF-GDPRMean

LF-GDPRMean [19] is a one-round protocol derived from the LF-GDPR framework for graph metric estimation under local differential privacy. The method supports customized privacy budget allocation between randomized response applied to adjacency vectors and Laplacian noise added to node degree perturbation. LF-GDPRMean uses the mean graph degree as a representative statistic to determine privacy budget allocation. For triangle counting, the collector first estimates incident triangles using the noisy adjacency matrix and subsequently calibrates estimates using perturbed node degrees to obtain an unbiased count. Since the method performs triangle counting over a dense perturbed graph, we parallelized this step in our implementation. In most cases (when $\epsilon < 5$), despite this parallelization, we do not observe an effect on the relative runtime ranking of the LF-GDPRMean algorithm.

TwoPhaseO

TwoPhaseO [15] is a two-round protocol designed to reduce Laplacian noise through extended local views. In phase one, participants estimate the maximum number of common neighbors shared with other nodes in their two-hop neighborhood. This information is combined with perturbed node degree in phase two to reduce the scale of Laplacian noise required for estimation. The protocol assumes that users have access to extended neighborhood information; for example, in a social network setting, users observe the friend lists of their friends.

VC

VC [9] is a three-round vertex-centric protocol that collects perturbed adjacency information and node degrees before estimating incident triangles within local neighborhoods. Unlike approaches that rely on exhaustive triangle counting over the noisy graph, VC restricts computation to local neighborhoods in order to reduce runtime. The protocol additionally proposes a privacy budget allocation strategy based on Newton’s method to optimize the split between randomized response and Laplacian perturbation. However, due to inconsistencies in the published derivation, we were unable to reproduce this optimization step. In our experiments, evenly splitting the privacy budget empirically yielded lower error.

4.2 Setup

All experiments were conducted on a 13th Gen Intel Core i9-13900K system with 32 threads and 128GB RAM. The algorithms are implemented with C++17. We store graph edges in a compact adjacency matrix representation, keeping memory usage low for sparse graphs which are commonly encountered in practical triangle counting applications.

We evaluate algorithms using both real-world and synthetic graph datasets. Real-world evaluations are conducted using the Enron email communication network, while synthetic evaluations employ the Erdős-Rényi random graph model to systematically analyze the effect of graph size and edge density on performance. Experiments vary privacy budget ϵ , number of nodes n , and edge existence probability p to characterize privacy-utility-runtime trade-offs across different graph conditions.

Across all experiments, we report algorithm performance with respect to three evaluation dimensions: Relative Error, runtime for computational efficiency, and communication overhead for practical deployment cost. PLUTO supports L2-loss as an additionally efficacy metric; however those results were not reported below as they closely mirrored the relative error metrics and only served to reinforce the conclusions drawn from the utility evaluation done using relative error. The results of each experiment are averaged over 10 runs to account for the randomness of privacy mechanisms.

All code, including experimental parameter files and random seeds, are provided in our publicly available repository².

4.3 Varying Privacy Budgets with Real-World Data

We first evaluate algorithm performance under varying privacy budgets ϵ using the Enron email communication network, a sparse real-world graph containing approximately half a million email interactions and $n = 36,692$ nodes. This experiment investigates how privacy strength influences utility, runtime, and communication overhead in realistic graph settings. All algorithms were evaluated with ϵ from 1 to 8. For algorithms with lower runtimes, we additionally consider ϵ values from 0.1 to 0.75.

Figure 2a presents the relative error across privacy budgets while Figure 2d presents the standard deviation of these experiments across all 10 runs of each experiment. VC [9] consistently achieves the lowest error across all values of ϵ , remaining below 0.1 throughout. TwoPhaseO [15] performs competitively, although it assumes users permit their neighbors to observe all local connections. ARROneNS [11] exhibits errors roughly two orders of magnitude larger than Local2Rounds [10], consistent with prior observations that communication-efficient edge sampling in Asymmetric Randomized Response introduces additional information loss. The resulting information loss increases estimation error. LFGDPRMean [19] produces substantially higher error than competing methods and only becomes competitive at sufficiently large privacy budgets ($\epsilon \geq 5$).

² <https://github.com/longg2610/PLUTO>

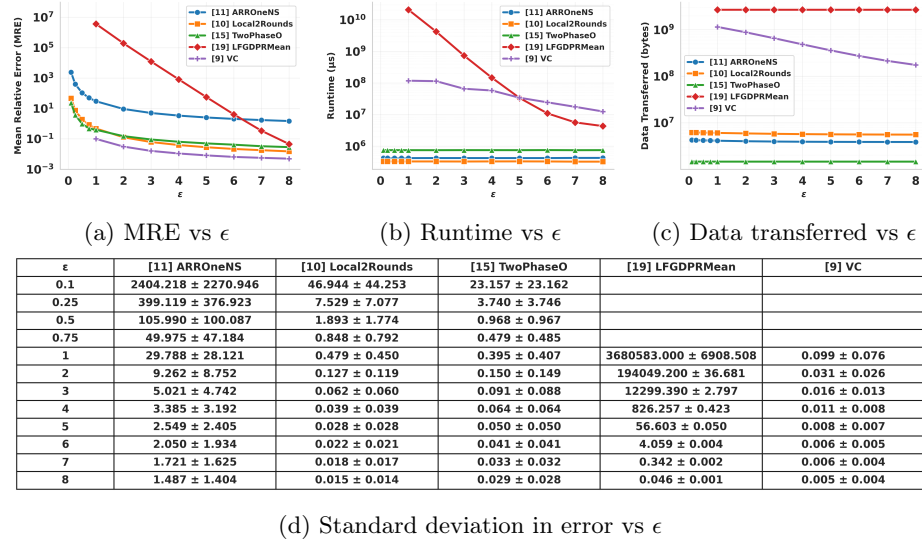


Fig. 2: Mean relative error, runtime (in μs), and communication overhead (in bytes) with privacy budget ϵ from 0.1 to 8, $n = 36692$.

We also note the relative stability of these results as captured in the standard deviation values reported in Figure 2d.

Figure 2b evaluates computational cost. LFGDPRMean incurs observably high computation because triangle counting is performed over the entire noisy graph. Under randomized response, smaller values of ϵ increase edge flipping probability, causing sparse graphs to become substantially denser after perturbation [10]. Since triangle enumeration must then operate over a denser graph, runtime increases dramatically. We note that our implementation of LFGDPRMean was parallelized to enable ϵ experiments, and that without this parallelization the runtime would continue to exceed VC when $\epsilon \geq 5$. LFGDPRMean was parallelized because it became prohibitively expensive for larger experimental settings. While the absolute runtimes are not directly comparable in this case the runtime conclusions on the cost of LFGDPRMean remain clear as it is definitively the most expensive algorithm.

In contrast, VC achieves substantially lower runtime despite also using noisy graph information. Rather than performing triangle counting over the entire perturbed graph, VC limits computation scope to the local neighborhood of each node taken from the original sparse graph requiring notably less noisy edge information compared with LFGDPRMean. TwoPhaseO similarly benefits from leveraging information from the real graph by counting triangles on the real graph, as evident from its low runtime. ARROneNS and Local2Rounds have a time complexity of $O(n^2 + nd_{max}^2)$, as their current implementations apply RR to only edges that are required at the second round [10].

Figure 2c presents communication overhead. LFGDPRMean incurs the highest communication cost (approximately 2.5GB) because each participant perturbs and transmits roughly half of an adjacency bit vector [19]. VC exhibits

decreasing communication overhead as ϵ increases, since denser noisy graphs at smaller ϵ values increase the likelihood of completing triangles and therefore increase transmissions. ARROneNS and Local2Rounds exhibit similar trends, although less pronounced. TwoPhaseO achieves the lowest communication overhead because it avoids transferring adjacency vectors entirely. Since communication variability depends primarily on participant selection during the second round, changes across privacy budgets remain comparatively small.

4.4 Varying Number of Nodes in Synthetic Graphs

We next evaluate scalability under varying graph size using both Erdős-Rényi [4] and Barabási-Albert [2] models. In these experiments, we fix $\epsilon = 1$, $p = 0.01$ in the case of the Erdős-Rényi model, and $m = 5$ in the case of Barabási-Albert while varying the number of nodes n . Due to the computational expense of LFGDPRMean [19], we restricted our evaluation to graphs for the algorithm to $n \leq 30000$. All others were evaluated for $n \leq 50000$

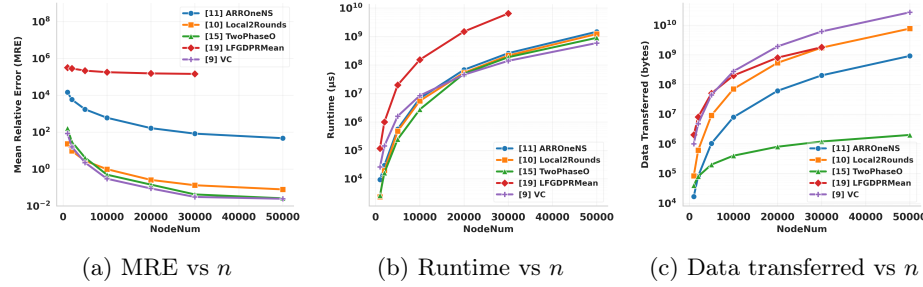


Fig. 3: Mean relative error, runtime (in μs), and communication overhead (in bytes) in Erdős-Rényi graphs with n from 0 to 50000, $p = 0.01$.

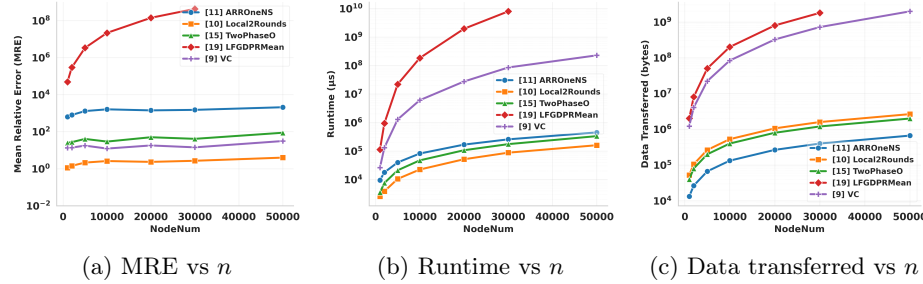


Fig. 4: Mean relative error, runtime (in μs), and communication overhead (in bytes) in Barabási-Albert graphs with n from 0 to 50000, $m = 5$.

Figures 3a and 4a presents the relative error. Accuracy trends remain largely consistent with those observed on the Enron graph. VC continues to achieve the lowest estimation error, remaining below 10^2 , while LFGDPRMean exhibits the highest error. More broadly, relative algorithm performance remains stable in the Erdős-Rényi results as graph size increases, suggesting that differences in estimation quality persist across larger synthetic settings. Notably, error increases

in the case of the LFGDPRMean algorithm as n increases in the Barabási-Albert results. Given the Erdős-Rényi approach has a fixed density controlled by the p parameter while the Barabási-Albert approach alternatively has a fixed edge count controlled by the m parameter, a consequence in the Barabási-Albert setting is that graphs get sparser as n grows while Erdős-Rényi maintain consistent density. This sparsity reacts particularly strongly to the noise from the LDP mechanism in the LFGDPRMean algorithm causing the rise in error seen in Figure 4a.

Runtime behavior, however, differs substantially from observations on the Enron graph in the Erdős-Rényi model (Figure 3b). While LFGDPRMean [19] remains the slowest algorithm, VC [9] achieves runtime comparable to ARROneNS [11], Local2Rounds [10], and TwoPhaseO [15], contrasting with Experiment 1 in section 4.3 and in the Barabási-Albert model in Figure 4b where VC was one to two orders of magnitude slower than these methods. Notably, VC runtime remains comparatively stable across graph settings, while ARROneNS, Local2Rounds, and TwoPhaseO experience substantial runtime growth as graph size increases. This suggests that VC does not consistently incur higher computational cost and that graph structure disproportionately affects the runtime of competing methods.

These results highlight an important observation: algorithm runtime depends not only on graph size, but also heavily on graph structure. The relative computational efficiency of methods can therefore vary substantially across graph topologies, even when node counts are similar.

Figures 3c and 4c present communication overhead. We first inspect the results of the Erdős-Rényi graphs. As expected, TwoPhaseO [15] transfers the least amount of data as the algorithm does not transfer neighbor lists while ARROneNS [11] is more communication-efficient than its predecessor Local2Rounds [10]. Interestingly, VC exceeds the communication cost of LFGDPRMean when $n \geq 10,000$, reversing the behavior observed on the Enron graph (Figure 2c).

We compare this results to the Barabási-Albert model where we see a clear separation between the two most communication expensive approaches and the other three. However, there is not as significant a separation between TwoPhaseO and ARROneNS or Local2Rounds as was seen with the Erdős-Rényi graphs. We again attribute this to the sparsity that is induced in the Barabási-Albert graphs as n increases with a fixed m . This presents another case wherein graph topology plays a clear role in algorithm deployment considerations.

4.5 Varying Edge Parameters in Synthetic Graphs

Our final experiments investigate the effect of edge parameters in synthetic graphs by varying edge existence probability p in the Erdős-Rényi model and the edge count m in the Barabási-Albert model while fixing $n = 10,000$ and $\epsilon = 1$.

Figures 5a and 6a presents the relative error. The accuracy trends with increasing p are consistent with those for increasing n in the Erdős-Rényi model. As p increases, estimation error decreases across methods, consistent with the broader observation that differential privacy methods benefit from larger effective data

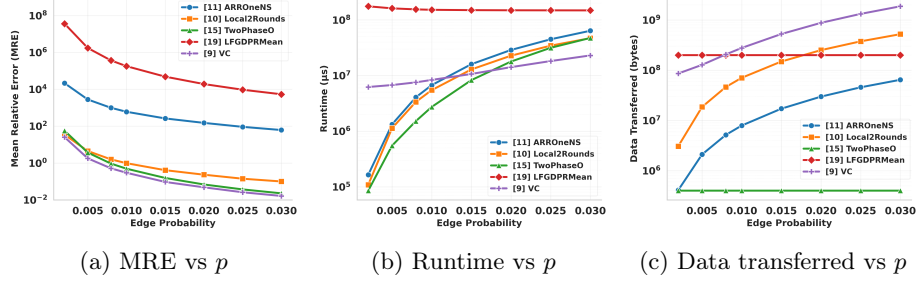


Fig. 5: Mean relative error, runtime (in μs), and communication overhead (in bytes) in Erdős-Rényi graphs with p from 0.002 to 0.030, $n = 10000$.

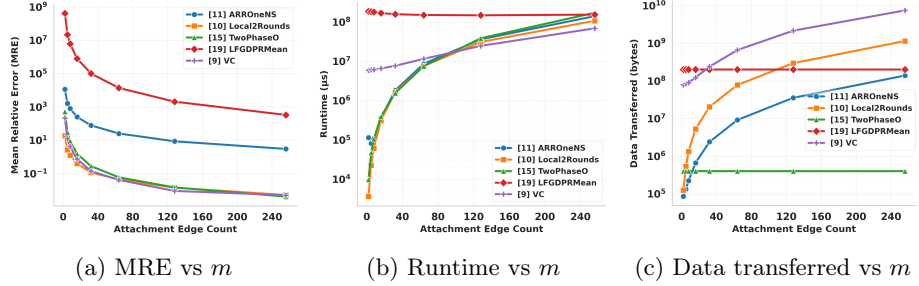


Fig. 6: Mean relative error, runtime (in μs), and communication overhead (in bytes) in Barabási-Albert graphs with m from 2 to 256, $n = 10000$.

size. VC continues to achieve the strongest accuracy, while LFGDPRMean remains the least accurate approach.

In the Barabási-Albert model, we see results similar to those in the Erdős-Rényi model. Notably, we do not see the increase in error observed in the experiments with increasing n using the LFGDPRMean algorithm as the increasing m while using a fixed n leads to a steady increase in graph density.

Figures 5b and 6b evaluate runtime sensitivity to graph density. LFGDPRMean [19] continues to exhibit the highest runtime ($> 10^8 \mu s$); however, we do observe a slight decrease in execution time as p increases. As the underlying graph becomes denser, randomized response under a fixed privacy budget ($\epsilon = 1$) produces comparatively sparser noisy graphs, reducing triangle counting cost. This mirrors the effect observed in Figure 2b when increasing ϵ , where higher privacy budgets similarly reduce graph densification.

The other algorithms evaluated observed different scaling behavior. While the execution time of AROneNS [11], Local2Rounds [10], and TwoPhaseO [15] increases substantially as graph density rises, ranging from approximately $10^5 \mu s$ to $10^8 \mu s$, VC [9] remains comparatively stable across values of p ($\approx 10^7 \mu s$), eventually achieving the lowest runtime out of all algorithms when $p \geq 0.02$. This again demonstrates that graph structure can substantially alter relative runtime rankings across methods.

Figures 5c and 6c present communication overhead. Since LFGDPRMean always transmits approximately half of an adjacency bit vector and TwoPhaseO does not transfer adjacency vectors, communication cost for these methods

depends primarily on the number of nodes rather than graph density. Consequently, communication remains relatively stable across values of p . Although LFGDPRMean exhibits the highest communication cost across privacy budgets in Experiment 1 (Figure 2c), it becomes less expensive than VC when $p \geq 0.01$ and Local2Rounds when $p \geq 0.02$. In the Barabási-Albert model, we again see results similar to those in the Erdős-Rényi model rather than the differences observed in the experiments with increasing n . These findings again emphasize that graph topology strongly influences algorithm performance and practical deployment cost.

4.6 Insights

Using PLUTO, we derive several practical insights regarding locally differentially private triangle counting algorithms.

Insight 1: Graph topology substantially influences algorithm behavior.

Across all experiments, algorithm rankings are not invariant to graph structure. Methods that perform favorably on sparse real-world networks may exhibit different runtime and communication characteristics on synthetic Erdős-Rényi and Barabási-Albert graphs, even under comparable graph sizes. These findings suggest that evaluation on a single dataset or graph family is insufficient for understanding practical performance.

Insight 2: Communication-efficient methods incur predictable utility trade-offs.

ARROneNS [11] and Local2Rounds [10] provide competitive trade-offs between accuracy and communication overhead, while ARROneNS [11] additionally enables tunable communication reduction. Across all three experiments, reducing communication overhead leads to increased error. However, the runtime difference between these two protocols remains mostly unchanged, indicating that the accuracy-communication tradeoff can be tuned independently of computational cost.

Insight 3: TwoPhaseO is effective under permissive privacy assumptions.

TwoPhaseO [15] is effective when exposing neighbor lists is acceptable and communication budgets are limited. TwoPhaseO [15] relies on a heuristic to determine the number of participants in the second round, which shows little variation as ϵ changes. Its transmission cost thus only scales with the total number of nodes in the graph (Figure 3c).

Insight 4: LFGDPRMean performs best under weaker privacy requirements.

LFGDPRMean [19] exhibits low accuracy and high runtime except at large ϵ , while incurring substantial communication overhead (Figure 2). This overhead however is only dependent on the number of nodes (Figure 3c). LFGDPRMean [19] counts triangles over the whole noisy graph, and thus is more effective when a larger privacy budget is available, as the noisy graph does not become overly dense at higher values of ϵ .

Insight 5: VC consistently achieves the strongest accuracy.

VC [9] achieves the highest accuracy among evaluated methods, although at relatively high communication cost. This overhead decreases as ϵ increases (Figure 2c). We counted VC [9]’s cost incurred when users report their noisy degrees used for Laplacian noise sensitivity in the final round. The extra round to report noisy degrees in VC [9] most likely explains the higher transmission cost compared to AROneNS [11] and Local2Rounds [10], which take the maximum degree in the graph as the global sensitivity, under the assumption that this statistic is public [11].

5 Related Work

Benchmarking frameworks for differentially private graph analytics have primarily focused on centralized privacy settings. DPGraph [18] evaluates selected subgraph counting and degree-sequence queries under edge-DP and node-DP, both of which assume a trusted data curator with access to the raw graph. The framework provides a web-based benchmarking platform with built-in algorithms, real-world graph datasets, and visual analysis tools for comparing privacy-utility trade-offs across privacy budgets and graph sizes. While DPGraph supports principled evaluation of centrally private graph analytics, it does not address the unique challenges introduced by local differential privacy, including distributed data ownership, multi-round communication, and perturbation applied independently by participants.

PGB [14] introduces the Private Graph Benchmark, a framework for evaluating differentially private synthetic graph generation methods. The system categorizes prior work according to privacy mechanisms, graph datasets, privacy models, and utility metrics, and evaluates representative approaches using both utility and computational analyses. Although PGB demonstrates the value of systematic benchmarking in privacy-preserving graph analysis, its focus is fundamentally different from ours: rather than graph analytic algorithms under standardized experimental conditions, PGB focuses on benchmarking synthetic graph generation quality.

Beyond graph analytics, several benchmarking frameworks have been proposed for tabular differentially private data analysis. DPBench [7] evaluates algorithms for answering one- and two-dimensional range queries, while DPComp [8] provides a publicly accessible platform for principled evaluation of private data analysis and facilitates dissemination of implementations, datasets, and experimental results. Collectively, these systems highlight the importance of reproducible benchmarking infrastructure for understanding privacy-utility trade-offs and enabling fair comparison across competing methods.

Despite these advances, existing benchmarking systems primarily focus on centralized graph analytics, synthetic graph generation, or tabular data analysis. Although prior LDP graph works commonly compare against existing baselines, to the best of our knowledge no unified benchmarking framework exists for reproducible and standardized evaluation of locally differentially private graph analytic algorithms, particularly triangle counting methods. PLUTO addresses this gap by providing a configurable and extensible framework for controlled

comparison across datasets, privacy budgets, graph structures, runtime, and communication overhead.

6 Conclusion

In this paper, we introduced PLUTO, a unified benchmarking framework for evaluating triangle counting algorithms under LDP. PLUTO addresses the lack of a standardized and reproducible evaluation infrastructure for LDP graph analytics by providing configurable support for datasets, privacy mechanisms, algorithm parameters, runtime measurement, and communication accounting.

Using PLUTO, we conducted a systematic evaluation of five state-of-the-art LDP triangle counting methods across varying privacy budgets, graph sizes, and graph topologies. Our results demonstrate that algorithm performance depends heavily on graph structure and experimental configuration, with substantial trade-offs emerging among utility, runtime, and communication overhead. In particular, our experiments reveal that relative algorithm rankings may change significantly across graph families, highlighting the importance of evaluation under diverse and controlled settings. The PLUTO system further supports future work studying real-world datasets such as GPlus and IMDB which are already supported directly within the PLUTO framework.

We additionally derived practical insights regarding the deployment characteristics of existing methods, including the effects of graph densification under randomized response, communication-efficient protocol design, and the impact of topology on computational scalability. As LDP graph analytics continue to evolve, we hope that PLUTO can serve as a reproducible experimental foundation for future research on graph statistics, privacy mechanisms, and communication-efficient LDP protocols. Future work includes extending PLUTO to additional graph analytics tasks beyond triangle counting and incorporating newly proposed locally private graph algorithms as they emerge.

References

1. Al Hasan, M., Dave, V.S.: Triangle counting in large networks: a review. *WIREs Data Mining and Knowledge Discovery* **8**(2), e1226 (2018). <https://doi.org/https://doi.org/10.1002/widm.1226>, <https://wires.onlinelibrary.wiley.com/doi/abs/10.1002/widm.1226>
2. Albert, R., Barabási, A.L.: Statistical mechanics of complex networks. *Reviews of Modern Physics* **74**(1), 47–97 (Jan 2002). <https://doi.org/10.1103/revmodphys.74.47>
3. Dwork, C., Roth, A.: The algorithmic foundations of differential privacy. *Found. Trends Theor. Comput. Sci.* **9**(3–4), 211–407 (Aug 2014). <https://doi.org/10.1561/04000000042>, <https://doi.org/10.1561/04000000042>
4. Erdős, P., Rényi, A., et al.: On the evolution of random graphs. *Publications of the* (1960)
5. Erlingsson, U., Pihur, V., Korolova, A.: Rappor: Randomized aggregatable privacy-preserving ordinal response. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security*. pp. 1054–1067. CCS ’14, ACM (Nov 2014). <https://doi.org/10.1145/2660267.2660348>

6. Gursay, M.E., Tamersoy, A., Truex, S., Wei, W., Liu, L.: Secure and utility-aware data collection with condensed local differential privacy. *IEEE Trans. Dependable Secur. Comput.* **18**(5), 2365–2378 (Sep 2021). <https://doi.org/10.1109/TDSC.2019.2949041>
7. Hay, M., Machanavajjhala, A., Miklau, G., Chen, Y., Zhang, D.: Principled evaluation of differentially private algorithms using dpbench (2015), <https://arxiv.org/abs/1512.04817>
8. Hay, M., Machanavajjhala, A., Miklau, G., Chen, Y., Zhang, D., Bissias, G.: Exploring privacy-accuracy tradeoffs using dpcomp. In: *Proceedings of the 2016 International Conference on Management of Data*. p. 2101–2104. SIGMOD '16, Association for Computing Machinery, New York, NY, USA (2016). <https://doi.org/10.1145/2882903.2899387>, <https://doi.org/10.1145/2882903.2899387>
9. He, Y., Wang, K., Zhang, W., Lin, X., Zhang, Y., Ni, W.: Robust privacy-preserving triangle counting under edge local differential privacy. *Proc. ACM Manag. Data* **3**(3) (Jun 2025). <https://doi.org/10.1145/3725348>
10. Imola, J., Murakami, T., Chaudhuri, K.: Locally differentially private analysis of graph statistics. In: *30th USENIX Security Symposium (USENIX Security 21)*. pp. 983–1000. USENIX Association (Aug 2021)
11. Imola, J., Murakami, T., Chaudhuri, K.: Communication-Efficient triangle counting under local differential privacy. In: *31st USENIX Security Symposium (USENIX Security 22)*. pp. 537–554. USENIX Association, Boston, MA (Aug 2022)
12. Leskovec, J., Krevl, A.: SNAP Datasets: Stanford large network dataset collection. <http://snap.stanford.edu/data> (Jun 2014)
13. Leskovec, J., McAuley, J.: Learning to discover social circles in ego networks. In: Pereira, F., Burges, C., Bottou, L., Weinberger, K. (eds.) *Advances in Neural Information Processing Systems*. vol. 25. Curran Associates, Inc. (2012)
14. Liu, S., Du, H., Cao, Y., Yan, B., Liu, J., Yoshikawa, M.: Pgb: Benchmarking differentially private synthetic graph generation algorithms (2024), <https://arxiv.org/abs/2408.02928>
15. Sun, H., Xiao, X., Khalil, I., Yang, Y., Qin, Z., Wang, H.W., Yu, T.: Analyzing sub-graph statistics from extended local views with decentralized differential privacy. In: *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. pp. 703–717. CCS '19, Association for Computing Machinery, New York, NY, USA (2019). <https://doi.org/10.1145/3319535.3354253>
16. Wang, T., Blocki, J., Li, N., Jha, S.: Locally differentially private protocols for frequency estimation. In: *26th USENIX Security Symposium (USENIX Security 17)*. pp. 729–745. USENIX Association, Vancouver, BC (Aug 2017)
17. Warner, S.L.: Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American statistical association* **60**(309), 63–69 (1965)
18. Xia, S., Chang, B., Knopf, K., He, Y., Tao, Y., He, X.: Dpgraph: A benchmark platform for differentially private graph analysis. In: *Proceedings of the 2021 International Conference on Management of Data*. p. 2808–2812. SIGMOD '21, Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3448016.3452756>, <https://doi.org/10.1145/3448016.3452756>
19. Ye, Q., Hu, H., Au, M.H., Meng, X., Xiao, X.: Lf-gdpr: A framework for estimating graph metrics with local differential privacy. *IEEE Transactions on Knowledge and Data Engineering* **34**(10), 4905–4920 (2022). <https://doi.org/10.1109/TKDE.2020.3047124>

Differentially Private Community Detection with Privacy Amplification by Shuffling

Xincheng Xu¹, M.A.P. Chamikara^{2*}, Thilina Ranbaduge^{2*}, Thierry Rakotoarivelo², David Smith², and Qing Wang¹

¹ Australian National University, Australia
{xincheng.xu, qing.wang}@anu.edu.au

² CSIRO, Australia

{chamikara.arachchige, thilina.ranbaduge, thierry.rakotoarivelo,
david.smith}@csiro.au

Abstract. Community detection is important in various application domains, as it helps uncover underlying patterns and structures in large graphs. Recent studies have applied differential privacy (DP) to protect sensitive information about communities, but existing techniques either rely on a trusted central party or introduce excessive noise. We propose *Shuffle-based Differentially Private Community Detection (SDPCD)*, a framework that uses privacy amplification via shuffling to achieve a favorable privacy-utility trade-off for community detection in undirected graphs. SDPCD uses a two-stage protocol: a first perturbation stage identifies community groups in a noisy graph, and a second independent perturbation stage produces the reports, which are shuffled and released. This separation enables a rigorous application of privacy amplification by shuffling. We prove two complementary guarantees: an unconditional pure $(\epsilon_{s1} + \epsilon_{s2}, 0)$ -Edge DP baseline and a refined $(\epsilon_{s1} + \epsilon_{\text{amp}}, \delta)$ -Edge DP bound obtained by proving a conditional Stage-2 DP guarantee for each fixed Stage-1 partition and then composing adaptively with Stage-1. Both guarantees are stated against the untrusted downstream analyst. Experiments on four real-world datasets show that SDPCD outperforms state-of-the-art differentially private community detection methods.

Keywords: Differential privacy · Community detection · Privacy amplification · Shuffle model · Graph analytics

1 Introduction

Graph analytics capture complex relationships among entities across domains such as social networks, transportation, genetics, and cybersecurity [14]. Community detection (i.e., identifying dense subgraphs within large networks) is central to these analyses, with applications ranging from protein function prediction to social group discovery [4].

* These authors contributed equally to this work.

Graph data often contain sensitive information such as political affiliations, medical conditions, or social connections that can be exposed during analysis. *Differential privacy (DP)* [10] has become a standard approach for protecting such data, with applications to graph statistics and privacy-preserving graph publication [21].

The privacy-utility gap. A fundamental challenge is to balance privacy and utility in community detection. *Central Differential Privacy (CDP)* requires a fully trusted curator who collects raw data and adds noise centrally; any compromise of this curator exposes all user data [19]. *Local Differential Privacy (LDP)* eliminates this single point of trust by having each user add noise locally before sharing, but this demands excessive noise because the error bound on aggregate statistics scales with the square root of the number of users [3]. In practice, LDP-based community detection algorithms introduce so many fictitious edges that the recovered communities bear little resemblance to the true structure at moderate privacy budgets.

Privacy amplification via shuffling. The *shuffle model* [5] interpolates between CDP and LDP: users apply local randomizers, then a trusted shuffler randomly permutes the reports before they reach the analyst. The anonymization induced by shuffling means that each report’s LDP guarantee is *amplified*, a global (ϵ, δ) -CDP guarantee can be achieved with $\epsilon \ll \epsilon_s$, where ϵ_s is the local budget [13, 2, 8, 12, 28]. This technique has been extensively studied in tabular settings but has not yet been applied to graph community detection, where the structure of connections must be preserved.

Our approach. We propose *Shuffle-based Differentially Private Community Detection (SDPCD)*, a two-stage protocol. In *Stage-1*, each user perturbs their adjacency list with ϵ_{s1} -Edge LDP noise; the honest-but-curious aggregator uses the resulting noisy graph to identify communities and define the shuffle partition. Since the partition is derived only from Stage-1 output, it is fully determined before Stage-2 begins. In *Stage-2*, each user independently applies fresh ϵ_{s2} -Edge LDP noise to their adjacency bits; the aggregator routes these fresh reports into the per-group shuffle lists, and each shuffler independently permutes one list. The minimum group size k ensures that every shuffle list is sufficiently large for amplification. Because Stage-2 reports are fresh, independent of Stage-1 randomness, the shuffle amplification theorem can then be applied to Stage-2 after fixing the Stage-1 partition and using the reduction developed in Section 5.3, and adaptive composition then accounts for Stage-1.

Contributions.

- 1) We propose SDPCD, the first framework combining community-aware grouping with shuffling for differentially private community detection, using a two-stage protocol that enables rigorous privacy amplification.
- 2) We prove two complementary privacy guarantees against the untrusted analyst:

- a *pure* $(\epsilon_{s1} + \epsilon_{s2}, 0)$ -Edge DP baseline (Theorem 2) that holds unconditionally for all k ;
- a *refined* $(\epsilon_{s1} + \epsilon_{\text{amp}}, \delta)$ -Edge DP guarantee (Theorem 3) obtained by proving that, for every fixed Stage-1 partition, the Stage-2 mechanism satisfies the amplified guarantee via a reduction to the single-message shuffle model, and then composing adaptively with Stage-1.

2 Related Work

2.1 Differential Privacy in Community Detection

CDP-based methods. Early approaches perturb graph representations at a trusted curator. Chen *et al.* [7] perturb dense adjacency subregions, while Xiao *et al.* [30] perturb edge probabilities in a Hierarchical Random Graph; both incur $O(n^2)$ cost. Nguyen *et al.* [24] later proposed the linear-cost Top- m Filter and the community-detection methods LouvainDP and ModDivisive [25].

LDP-based methods. LDPGen [26] generates synthetic graphs by grouping users according to perturbed degrees, while CGGen [18] additionally models intra- and inter-group correlations. CPRR [16] incorporates community-aware randomized response sampling but is restricted to sparse graphs. LF-GDPR [31] estimates modularity from perturbed adjacency and degree information, while LDPCD [32] combines local perturbation with extremal optimization [9].

Positioning of SDPCD. SDPCD operates in the shuffle model between CDP and LDP. Unlike CDP methods, it does not require a fully trusted data collector; unlike LDP methods, it injects substantially less noise by exploiting shuffling amplification. Unlike CPRR, it is not restricted to sparse graphs, and unlike LDPGen, it preserves community structure because grouping is based on graph topology rather than degree alone.

2.2 Privacy Amplification by Shuffling

Shuffling was introduced by Prochlo [5] to decouple individual reports from their source, providing anonymization. Erlingsson *et al.* [12] formalized the privacy amplification guarantee; Balle *et al.* [2] and Cheu *et al.* [8] strengthened the analysis; Feldman *et al.* [13] obtained the tightest known bounds for the single-message shuffle model.

For graph-structured data, Imola *et al.* [15] applied *wedge shuffling* to count triangles and 4-cycles, exploiting the local structure of wedges; however, this technique is specific to subgraph counting and does not extend to community detection. Abouei and Canonne [1] proposed group shuffling for tabular data, randomly partitioning users into groups before shuffling; applied to graphs, this loses connectivity information. SDPCD differs in that it groups users according to their estimated community membership, thereby preserving the edge structure needed for accurate community detection, and in that it separates the grouping stage from the shuffled release, enabling rigorous amplification.

Table 1. Basic notation.

$G = (V, E)$	Undirected graph	A	Adjacency matrix
B_v	Local neighbouring list of v	$n = V $	Number of nodes
$\mathcal{C} = \{C_1, \dots, C_r\}$	Communities	q	Modularity
ϵ_{s1}	Stage-1 privacy budget	ϵ_{s2}	Stage-2 local budget
ϵ_{amp}	Amplified Stage-2 budget	δ	Failure probability
k	Minimum group size	m	Minimum shuffle list size
$\mathcal{M}_{\text{RABV}}$	RABV mechanism	$\hat{A}$	Shuffled adjacency matrix

3 Preliminaries

This section introduces the graph, differential privacy, and privacy amplification by shuffling concepts used throughout the paper. Table 1 summarizes the main notation.

3.1 Graph and Community Detection

An undirected graph $G=(V, E)$ has adjacency matrix A where $A_{uv}=1$ iff $(u, v) \in E$ and $A_{uv}=A_{vu}$. The local neighboring list B_v is row v of A ; since G is undirected, each undirected edge (u, v) corresponds to the symmetric pair (A_{uv}, A_{vu}) . We focus on the *global* community-detection task: partition V into r non-overlapping communities $\mathcal{C} = \{C_1, \dots, C_r\}$ such that intra-community edges are dense and inter-community edges are sparse. Quality is measured by *modularity* [23]:

$$q = \sum_{i=1}^r \left[\frac{\ell_{C_i}}{|E|} - \left(\frac{d_{C_i}}{2|E|} \right)^2 \right], \quad q \in [-1, 1], \quad (1)$$

where ℓ_{C_i} is the intra-community edge count and d_{C_i} is the sum of node degrees in C_i ; higher q indicates a better partition.

3.2 Differential Privacy

Definition 1 ((ϵ, δ)-DP [10]). A randomized algorithm $\mathcal{M}$ is (ϵ, δ) -differentially private if for all neighboring databases D, D' differing in one record, and all measurable sets $\mathcal{S}$: $\Pr[\mathcal{M}(D) \in \mathcal{S}] \leq e^\epsilon \Pr[\mathcal{M}(D') \in \mathcal{S}] + \delta$. When $\delta = 0$ we say $\mathcal{M}$ is purely ϵ -DP.

Definition 2 (ϵ -LDP [19]). A local algorithm $\mathcal{R}$ is ϵ -locally differentially private if for all input pairs x, x' and all measurable $\mathcal{S}$: $\Pr[\mathcal{R}(x) \in \mathcal{S}] \leq e^\epsilon \Pr[\mathcal{R}(x') \in \mathcal{S}]$.

Definition 3 (Edge CDP / Edge LDP [21, 26]). A randomized algorithm $\mathcal{M}$ is (ϵ, δ) -Edge CDP if for all neighboring undirected graphs G, G' differing in exactly one edge, and all measurable sets $\mathcal{S}$: $\Pr[\mathcal{M}(G) \in \mathcal{S}] \leq e^\epsilon \Pr[\mathcal{M}(G') \in \mathcal{S}] + \delta$. A local randomized algorithm $\mathcal{R}$ is ϵ -Edge LDP if for all neighboring local lists B_v, B'_v differing in exactly one adjacency bit, and all measurable sets $\mathcal{S}$: $\Pr[\mathcal{R}(B_v) \in \mathcal{S}] \leq e^\epsilon \Pr[\mathcal{R}(B'_v) \in \mathcal{S}]$.

Definition 4 (Warner’s Randomized Response (RR) [29]). For a binary input $b \in \{0, 1\}$, randomized response $\mathcal{R}_\epsilon$ outputs b with probability $\Pr[\mathcal{R}_\epsilon(b)=b] = \frac{e^\epsilon}{e^\epsilon+1}$ and $1-b$ with probability $\Pr[\mathcal{R}_\epsilon(b)=1-b] = \frac{1}{e^\epsilon+1}$.

We also rely on three standard DP composition properties.

Lemma 1 (Parallel Composition [22]). Let $\mathcal{M}$ be ϵ -DP and $D = D_1 \sqcup \dots \sqcup D_m$ (disjoint). Releasing $\{\mathcal{M}(D_i)\}_{i=1}^m$ simultaneously satisfies ϵ -DP.

Lemma 2 (Post-Processing [11]). If $\mathcal{M}$ is (ϵ, δ) -DP, then $\mathcal{H} \circ \mathcal{M}$ is (ϵ, δ) -DP for any (possibly randomized) function $\mathcal{H}$ independent of the raw data.

Lemma 3 (Adaptive Composition [10]). Let $\mathcal{M}_1$ be (ϵ_1, δ_1) -DP. Let $\mathcal{M}_2(\cdot; y)$ be (ϵ_2, δ_2) -DP for every fixed value y . Then the mechanism $D \mapsto (\mathcal{M}_1(D), \mathcal{M}_2(D; \mathcal{M}_1(D)))$ is $(\epsilon_1 + \epsilon_2, \delta_1 + \delta_2)$ -DP.

3.3 Privacy Amplification by Shuffling

The shuffle model places a trusted shuffler between users and the analyst. Users apply local randomizers; the shuffler uniformly permutes the resulting reports before forwarding them. The permutation breaks the link between a report and its originating user, providing anonymization that amplifies each user’s local privacy guarantee into a stronger central guarantee.

Theorem 1 (Privacy Amplification by Shuffling [13]). Let $\mathcal{R}$ be an ϵ_s -LDP randomizer and let $\mathcal{M}_s$ be the mechanism that applies $\mathcal{R}$ to each of n inputs and outputs them in uniformly random order. For any $\delta \in (0, 1)$, if ϵ_s satisfies

$$\epsilon_s \leq \log\left(\frac{n}{16 \log(2/\delta)}\right), \quad (2)$$

then $\mathcal{M}_s$ satisfies (ϵ, δ) -DP where

$$\epsilon = \log\left(1 + \frac{e^{\epsilon_s} - 1}{e^{\epsilon_s} + 1} \left(\frac{8\sqrt{e^{\epsilon_s} \log(4/\delta)}}{\sqrt{n}} + \frac{8e^{\epsilon_s}}{n} \right)\right). \quad (3)$$

Intuitively, ϵ in (3) decreases as n grows, so larger shuffle lists yield stronger amplification. This is the key property we exploit in SDPCD.

4 SDPCD Framework

SDPCD (Figure 1) involves four entities: *users* (graph nodes) who hold sensitive neighboring lists; an *honest-but-curious aggregator* who orchestrates the protocol and executes it faithfully but may inspect its intermediate inputs (e.g., a head office among the users); *shufflers* who are also honest-but-curious; and an *untrusted analyst* who receives only the shuffled adjacency matrix $\hat{A}$.

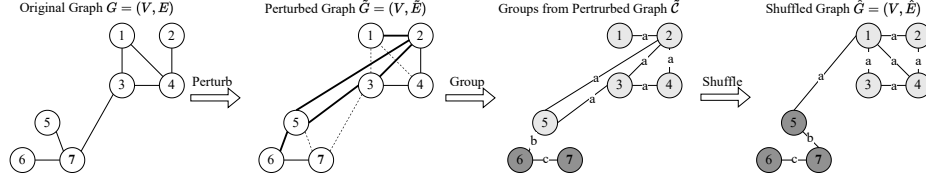


Fig. 1. SDPCD overview. **Stage-1: Grouping:** each user applies ϵ_{s1} -Edge LDP noise to their adjacency list; the aggregator forms $\tilde{A}^{(1)}$, runs community detection, and refines groups to enforce minimum size k . **Stage-2: Shuffle:** each user independently applies fresh ϵ_{s2} -Edge LDP noise; the aggregator routes these fresh perturbed bits into per-group shuffle lists; each shuffler independently permutes one list; the aggregator reconstructs $\hat{A}$ for the (untrusted) analyst.

The privacy guarantee is stated *against the analyst*: the analyst learns only $\hat{A}$ and any post-processing thereof. The aggregator observes both the Stage-1 noisy matrix $\tilde{A}^{(1)}$ and the pre-shuffle Stage-2 perturbed bits needed for routing; we do not claim the amplified guarantee against the aggregator.

The protocol proceeds in two stages. **Stage-1 (Grouping)**: each user perturbs their adjacency list with ϵ_{s1} -Edge LDP noise and sends it to the aggregator, who combines the perturbed bits into $\tilde{A}^{(1)}$, identifies communities via Algorithm 1, and fixes the shuffle partition $\mathcal{C}$. **Stage-2 (Shuffle)**: independently of Stage-1, each user perturbs their adjacency list again with fresh ϵ_{s2} -Edge LDP noise and sends the new perturbed bits to the aggregator, who routes them into per-group shuffle lists according to $\mathcal{C}$; each shuffler permutes one list; the aggregator reassembles $\hat{A}$.

The two stages use *independent* randomness. Crucially, the partition $\mathcal{C}$ is fixed before Stage-2 begins and depends only on Stage-1 randomness, not on the Stage-2 reports. Conditioned on any fixed Stage-1 outcome, the Stage-2 mechanism is therefore a shuffled local-randomization mechanism with a fixed, pre-determined partition. After the reduction from undirected edges to Stage-2 reports described in Section 5.3, this matches the setting of Theorem 1, and adaptive composition (Lemma 3) then combines the two stages.

4.1 Step 1: Perturbation via RABV

Both stages use the same local perturbation mechanism, *Randomized Adjacency Bit Vector (RABV)* [31], applied with independent randomness. Each application of RABV by user v perturbs exactly t consecutive bits starting from position $v+1 \bmod n$, where

$$t = \begin{cases} \lfloor n/2 \rfloor & \text{if } v \leq \lfloor n/2 \rfloor, \\ \lfloor (n-1)/2 \rfloor & \text{otherwise.} \end{cases}$$

This partition ensures that each pair of symmetric bits (A_{uv}, A_{vu}) is contributed by exactly one user, and the aggregator fills in the symmetric counterpart. Figure 2 illustrates the pattern.

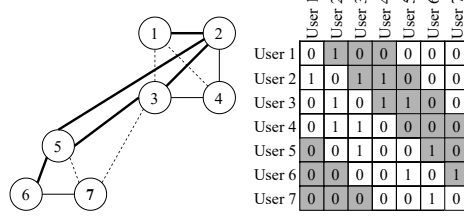


Fig. 2. RABV perturbation pattern for $n=5$ users. Grey blocks indicate which bits each user perturbs and uploads. Each pair of symmetric matrix entries is covered exactly once, ensuring ϵ_s -Edge LDP without double-counting.

Algorithm 1: Grouping Strategy

Input: Noisy adjacency matrix $\tilde{A}^{(1)}$; minimum group size k
Output: Communities $\mathcal{C}$

- 1 $\mathcal{C}_{tmp} \leftarrow \text{LOUVAIN}(\tilde{A}^{(1)});$ // initialize via modularity optimization
- 2 **while true do**
- 3 Compute feature matrix F : row i = mean of neighbour lists in $\mathcal{C}_{tmp,i}$;
- 4 Compute cosine similarity matrix S from F ;
- 5 $(i^*, j^*) \leftarrow \arg \max_{i < j} \{S[i][j] : |\mathcal{C}_{tmp,i}| < k \text{ or } |\mathcal{C}_{tmp,j}| < k\};$
- 6 **if no such pair exists then break;**
- 7 Merge $\mathcal{C}_{tmp,i^*}$ and $\mathcal{C}_{tmp,j^*}$;
- 8 **return** $\mathcal{C}_{tmp}$;

Lemma 4. *RABV with budget ϵ_s satisfies ϵ_s -Edge LDP.*

Proof. Each undirected edge is perturbed by exactly one user using ϵ_s -LDP Warner's RR. By Lemma 1 (parallel composition over disjoint bit sets), the perturbed bits jointly satisfy ϵ_s -Edge LDP. Combining them into a noisy matrix is post-processing (Lemma 2).

4.2 Step 2: Group Identification

Algorithm 1 is applied only to the Stage-1 noisy matrix $\tilde{A}^{(1)}$ and consists of two sub-stages. **Initialization:** Louvain [6] is applied to $\tilde{A}^{(1)}$ to obtain an initial partition that maximizes modularity. **Refinement:** communities smaller than k are iteratively merged with their most similar neighbor (by cosine similarity of mean neighbor-list vectors) until all groups satisfy $|C_i| \geq k$. This similarity-based merging strategy is analogous to centroid linkage in hierarchical clustering [17]. The resulting partition $\mathcal{C}$ is *fixed before Stage-2*, so it is independent of the Stage-2 reports.

Complexity. Louvain costs $O(|\tilde{E}| \log n)$; refinement costs $O(n^2)$. Space: $O(n^2)$.

Algorithm 2: Shuffling Step

Input: Stage-2 noisy matrix $\tilde{A}^{(2)}$; communities $\mathcal{C}$ (from Stage-1)
Output: Shuffled adjacency matrix $\hat{A}$

- 1 **for** each $C_i \in \mathcal{C}$ // intra-group lists do
- 2 $Sin_i \leftarrow \{\tilde{A}_{uv}^{(2)} : u < v, u, v \in C_i\}$; shuffle Sin_i uniformly at random;
- 3 **for** each pair (i, j) with $i < j$ // inter-group lists do
- 4 $Sout_{ij} \leftarrow \{\tilde{A}_{uv}^{(2)} : u \in C_i, v \in C_j\}$; shuffle $Sout_{ij}$ uniformly at random;
- 5 Reconstruct $\hat{A}$ by mapping each shuffled list back to its positions;
- 6 **return** $\hat{A}$;

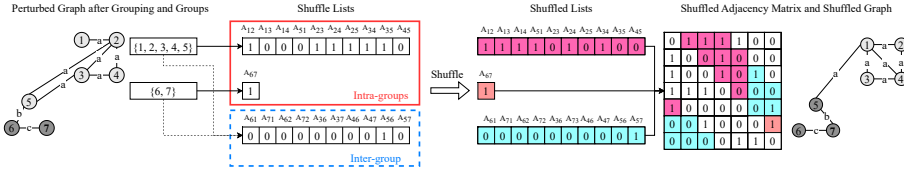


Fig. 3. Illustration of the shuffling step applied to Stage-2 noisy bits. Intra-group lists (red solid boxes) contain edges within each community; inter-group lists (blue dashed boxes) contain edges between communities. Each list is independently shuffled, preserving aggregate density while breaking the link between individual edge bits and their positions.

4.3 Step 3: Shuffling

Algorithm 2 and Figure 3 illustrate the shuffling step, which operates on the Stage-2 noisy matrix $\tilde{A}^{(2)}$. The communities from Algorithm 1 partition adjacency bits into shuffle lists:

- *Intra-group lists* Sin_i : bits for edges within C_i , of size $\binom{|C_i|}{2} \geq \binom{k}{2} = \frac{k(k-1)}{2}$.
- *Inter-group lists* $Sout_{ij}$: bits for edges between C_i and C_j , of size $|C_i| \cdot |C_j| \geq k^2 \geq \frac{k(k-1)}{2}$.

The minimum shuffle list size is therefore $m = \frac{k(k-1)}{2}$. Each list is shuffled independently by one shuffler. Since the lists partition the adjacency bits and each list contains Stage-2 LDP reports, parallel composition applies. After shuffling, the aggregator reconstructs $\hat{A}$ deterministically (post-processing).

Complexity. $O(n^2)$ time and space.

5 Privacy Analysis

We prove two complementary privacy guarantees for SDPCD, both at the edge level and both *against the untrusted analyst* who observes only $\hat{A}$.

5.1 Threat Model

Each node v locally applies RABV twice using independent randomness and sends both outputs to the *honest-but-curious aggregator*, which follows the protocol faithfully but may inspect its intermediate state. The *shufflers* are also honest-but-curious and handle only Stage-2 reports, while the *untrusted analyst* observes only $\hat{A}$ and any downstream function thereof.

Aggregator view. In Stage-1, the aggregator observes $\tilde{A}^{(1)}$ and the derived partition $\mathcal{C}$. In Stage-2, it also observes the pre-shuffle perturbed reports required for routing. Hence, the aggregator does not benefit from shuffle amplification, which is claimed only against the downstream analyst.

Privacy unit. A single undirected edge (u, v) .

5.2 Baseline Edge DP Guarantee

Theorem 2 (Baseline privacy of SDPCD). *SDPCD satisfies pure $(\epsilon_{s1} + \epsilon_{s2}, 0)$ -Edge DP against the analyst, regardless of the group size k .*

Proof. Let $\mathcal{M}_1 : G \mapsto \tilde{A}^{(1)}$ denote the Stage-1 RABV mechanism and $\mathcal{M}_2(G; \mathcal{C}) : G \mapsto \hat{A}$ denote the Stage-2 mechanism (fresh RABV followed by the deterministic routing-and-shuffling pipeline with fixed $\mathcal{C}$). By Lemma 4, $\mathcal{M}_1$ is ϵ_{s1} -Edge DP. For any fixed partition $\mathcal{C}$, Stage-2 applies fresh RABV at budget ϵ_{s2} to the raw graph; by Lemma 4 and post-processing (Lemma 2), $\mathcal{M}_2(\cdot; \mathcal{C})$ is ϵ_{s2} -Edge DP for every fixed $\mathcal{C}$. Applying Lemma 3 (adaptive composition with $\delta_1 = \delta_2 = 0$) to the joint transcript $(\tilde{A}^{(1)}, \hat{A})$ gives $(\epsilon_{s1} + \epsilon_{s2}, 0)$ -Edge DP; since the analyst observes only $\hat{A}$, the same bound holds by post-processing (Lemma 2).

5.3 Refined Amplified Guarantee via Shuffling

Because Stage-2 reports are generated by a fresh application of RABV, independent of Stage-1 randomness, for every fixed Stage-1 outcome, the Stage-2 mechanism is a shuffled local-randomization mechanism with a fixed, pre-determined partition. This matches the setting of Theorem 1 after the reduction from undirected edges to Stage-2 reports.

Lemma 5 (Stage-2 amplification). *Suppose the minimum group size satisfies $k \geq 2$ and ϵ_{s2} satisfies*

$$\epsilon_{s2} \leq \log \left(\frac{k(k-1)}{32 \log(2/\delta)} \right) \quad (4)$$

for some $\delta \in (0, 1)$. Then, for any fixed partition $\mathcal{C}$ with $|C_i| \geq k$ for all i , the Stage-2 mechanism $\mathcal{M}_2(\cdot; \mathcal{C})$ satisfies $(\epsilon_{\text{amp}}, \delta)$ -Edge DP, where

$$\epsilon_{\text{amp}} = \log \left(1 + \frac{e^{\epsilon_{s2}} - 1}{e^{\epsilon_{s2}} + 1} \left(\frac{8\sqrt{2e^{\epsilon_{s2}} \log(4/\delta)}}{\sqrt{k(k-1)}} + \frac{16e^{\epsilon_{s2}}}{k(k-1)} \right) \right). \quad (5)$$

Proof. Reduction to the single-message shuffle model. Under RABV, each undirected edge (u, v) is perturbed by exactly one user (the one whose RABV responsibility covers that pair), producing exactly one Stage-2 randomized report $\tilde{A}_{uv}^{(2)} \in \{0, 1\}$ drawn from an ϵ_{s2} -LDP randomizer. The partition $\mathcal{C}$, fixed by Stage-1 before Stage-2 begins, assigns each undirected edge to exactly one shuffle list in a deterministic, bijective manner that depends only on $\mathcal{C}$ and not on the Stage-2 reports themselves. Specifically, edge (u, v) maps to the intra-group list Sin_i if $u, v \in C_i$, and to the inter-group list $Sout_{ij}$ if $u \in C_i, v \in C_j$. Since each edge contributes exactly one report to exactly one list, and the lists are disjoint, we can therefore decompose the Stage-2 mechanism into shuffled sub-mechanisms over disjoint sets of edge reports, one per list. The reconstruction of $\hat{A}$ from the shuffled lists is deterministic given $\mathcal{C}$, and constitutes post-processing.

Applying the amplification theorem. Conditioned on any fixed Stage-1 outcome (and hence a fixed $\mathcal{C}$), each shuffle list contains fresh ϵ_{s2} -LDP reports from disjoint edges, and the partition is pre-determined. After the reduction above, each list matches the hypothesis of Theorem 1: an ϵ_{s2} -LDP randomizer applied to n disjoint edge-reports, with outputs uniformly permuted. Every shuffle list has size $\geq m = \frac{k(k-1)}{2}$ (intra-group: $\binom{|C_i|}{2} \geq \binom{k}{2}$; inter-group: $|C_i||C_j| \geq k^2 \geq m$). Applying Theorem 1 to each list with $n=m$ (a conservative lower bound; larger lists yield smaller ϵ) gives per-list $(\epsilon_{\text{amp}}, \delta)$ -Edge DP with (5) subject to (4).

Composition across lists. For fixed $\mathcal{C}$, the Stage-2 mechanism factorizes over the disjoint edge sets induced by the lists. Since neighboring graphs G, G' differ in exactly one edge, only one list's distribution changes; all other lists are identically distributed under G and G' . The joint release therefore inherits the same $(\epsilon_{\text{amp}}, \delta)$ -Edge DP bound by Lemma 1 (parallel composition over disjoint edge sets). The deterministic reconstruction of $\hat{A}$ from the shuffled lists is post-processing (Lemma 2).

Theorem 3 (Refined amplified privacy of SDPCD). *Under constraint (4), SDPCD satisfies $(\epsilon_{s1} + \epsilon_{\text{amp}}, \delta)$ -Edge DP against the analyst, where ϵ_{amp} is given by (5).*

Proof. Stage-1 ($\mathcal{M}_1$) is $(\epsilon_{s1}, 0)$ -Edge DP by Lemma 4 and post-processing. By Lemma 5, for every fixed Stage-1 outcome (and hence every fixed partition $\mathcal{C}$), the Stage-2 mechanism $\mathcal{M}_2(\cdot; \mathcal{C})$ satisfies $(\epsilon_{\text{amp}}, \delta)$ -Edge DP. Since Stage-1 and Stage-2 use independent randomness, Lemma 3 (adaptive composition) applies with $\epsilon_1 = \epsilon_{s1}$, $\delta_1 = 0$, $\epsilon_2 = \epsilon_{\text{amp}}$, $\delta_2 = \delta$, giving $(\epsilon_{s1} + \epsilon_{\text{amp}}, \delta)$ -Edge DP for the joint output $(\tilde{A}^{(1)}, \hat{A})$. Since the analyst observes only $\hat{A}$, post-processing (Lemma 2) gives the same guarantee for $G \mapsto \hat{A}$ alone.

Remark 1 (Baseline and amplified guarantees). Theorems 2 and 3 provide two privacy bounds for the analyst's view. SDPCD satisfies *both* Theorem 2 (pure $(\epsilon_{s1} + \epsilon_{s2}, 0)$ -Edge DP, no constraint on k) and Theorem 3 (approximate $(\epsilon_{s1} + \epsilon_{\text{amp}}, \delta)$ -Edge DP when constraint (4) holds and $\epsilon_{\text{amp}} \ll \epsilon_{s2}$). The amplified guarantee is strictly tighter when $\epsilon_{\text{amp}} + \epsilon_{s1} < \epsilon_{s1} + \epsilon_{s2}$, i.e. when $\epsilon_{\text{amp}} < \epsilon_{s2}$.

Remark 2 (Comparison with the single-stage approach). A simpler protocol would use a single RABV application for both grouping and shuffling. However, in that case, the partition is derived from the *same* randomized reports that are subsequently shuffled: conditioning on the partition fixes those report values, leaving only a permutation of an already-realized multiset. Standard shuffle amplification theorems require a *fresh* local-randomization stage after the partition is fixed; they do not apply to a permutation of pre-realized reports. The two-stage protocol resolves this by strictly separating the randomness used for grouping from the randomness used in the shuffled release.

6 Experimental Evaluation

6.1 Setup

Datasets (Table 2): four real-world undirected graphs from the SNAP repository [20]. *Facebook*: ego-networks from Facebook, edges are friendships. *Government*: mutual-like graph of government Facebook pages. *AstroPh*: co-authorship network from arXiv Astrophysics papers. *Enron*: email communication graph where an edge indicates at least one email exchanged. For evaluation, we apply a non-private community detection algorithm to each original, unperturbed graph and use the resulting communities as the non-private communities for comparison.

Baselines:

- RABV [31]: LDP perturbation of adjacency matrix followed by direct Louvain community detection and no shuffling.
- LDPGen [26]: degree-based LDP grouping and synthetic graph generation.
- LFGDPR [31]: LDP framework estimating modularity from perturbed adjacency and degree vectors.
- LouvainDP [25]: CDP method based on noisy Louvain on a compressed supergraph.

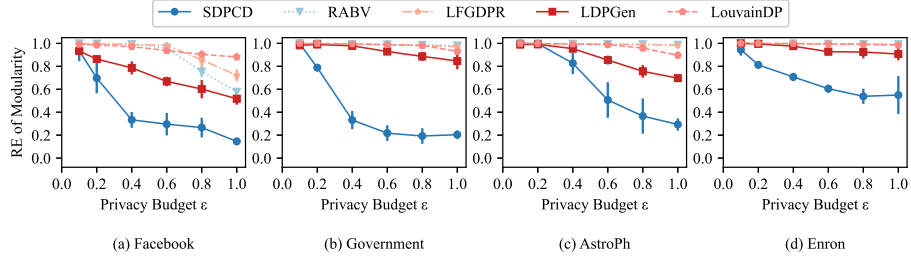
Metrics:

- *Relative Error (RE)* of modularity: $RE = |q - \tilde{q}|/q$, where q is the original graph’s modularity and $\tilde{q}$ is obtained from $\tilde{A}$. Lower is better.
- *Adjusted Rand Index (ARI)* and *Adjusted Mutual Information (AMI)* [27]: measure agreement between detected and non-private communities; higher is better.

Parameter setting: $\delta = 10^{-5}$; total privacy budget $\epsilon_{\text{total}} \in \{0.1, 0.2, 0.4, 0.6, 0.8, 1.0\}$ against the analyst. We split the budget equally: $\epsilon_{s1} = \epsilon_{\text{total}}/2$ for Stage-1, and choose ϵ_{s2} such that the amplified Stage-2 guarantee satisfies $\epsilon_{\text{amp}} = \epsilon_{\text{total}}/2$ by inverting (5) numerically subject to constraint (4). The total guarantee is then $(\epsilon_{s1} + \epsilon_{\text{amp}}, \delta) = (\epsilon_{\text{total}}, \delta)$ -Edge DP. Minimum group sizes are $k \in \{300, 600, 1200, 2400\}$ for Facebook, Government, AstroPh, Enron, respectively (chosen by the parameter study in Section 6.4). Each experiment is repeated 10 times; we report mean $\pm$ standard deviation. Code and data: <https://github.com/CharlieX001/SDPCD>.

Table 2. Dataset characteristics.

Dataset	$ V $	$ E $	$ C $
<i>Facebook</i>	4,039	88,234	17
<i>Government</i>	7,057	89,455	29
<i>AstroPh</i>	18,772	198,110	334
<i>Enron</i>	36,692	183,831	1,363

**Fig. 4.** Relative Error (RE) of modularity for all methods. SDPCD achieves the lowest RE for $\epsilon_{\text{total}} > 0.1$ on all four datasets, with the gap growing as ϵ_{total} increases.

6.2 Utility Evaluation

Modularity RE (Figure 4): SDPCD outperforms all baselines for $\epsilon_{\text{total}} > 0.1$ on all datasets. The advantage grows rapidly with ϵ_{total} : at $\epsilon_{\text{total}} = 1.0$, SDPCD’s RE is roughly half that of RABV on Facebook and Government. At $\epsilon_{\text{total}} = 0.1$, heavy noise causes all methods to converge to similar (poor) performance because the noisy graph loses most of its structural signal.

ARI and AMI (Figures 5-6): SDPCD’s detected communities are consistently most similar to the non-private communities. The advantage is most pronounced on Facebook and Government (smaller, denser datasets), where the Stage-1 grouping step can accurately identify the non-private community boundaries even in the noisy graph. On AstroPh and Enron (with hundreds and thousands of communities, respectively), all methods struggle at low ϵ_{total} , but SDPCD recovers more quickly as privacy relaxes.

Runtime (Table 3): At $\epsilon_{\text{total}} = 1$, SDPCD is faster than RABV and LFGDPR. LFGDPR is the slowest because its modularity gain estimation involves repeated community detection passes on the noisy graph. Although the two-stage protocol doubles the perturbation overhead per user, SDPCD benefits from a smaller ϵ_{s2} (relative to the total budget) and consequently a sparser noisy Stage-2 graph, leading to faster Louvain convergence. LouvainDP is fastest because it collapses the graph into supernodes before applying noise.

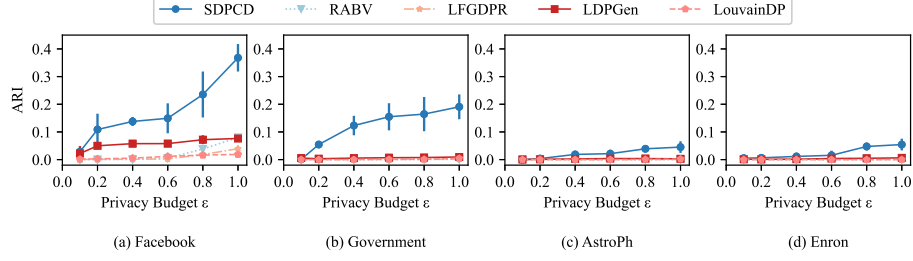


Fig. 5. Adjusted Rand Index (ARI). SDPCD consistently yields communities that are closest to the non-private communities for $\epsilon_{\text{total}} > 0.1$.

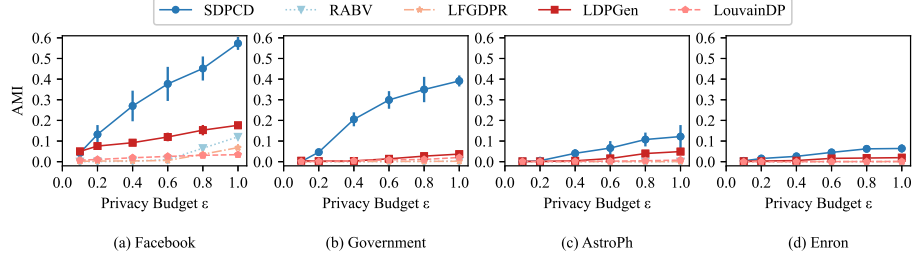


Fig. 6. Adjusted Mutual Information (AMI). Results corroborate the ARI findings.

6.3 Ablation Study

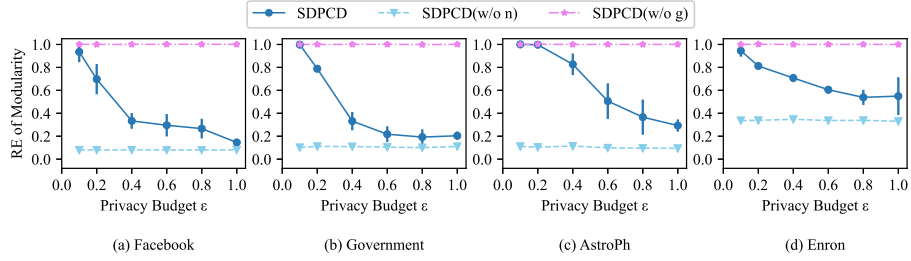
We isolate the contribution of the grouping step by comparing SDPCD against two ablations.

SDPCD (w/o g): all Stage-2 edges placed in a single global shuffle list (no community-based grouping). This maximizes the shuffle list size (strongest amplification) but completely destroys community structure by globally permuting all edges. Figure 7 confirms that $RE \rightarrow 1$ as ϵ_{total} grows, because the shuffled graph has no community structure.

SDPCD (w/o n): grouping and shuffling applied to the original A , without noise injection. This is not privacy-preserving but serves as an upper bound on what grouping and shuffling alone can achieve. On Facebook and Government, the introduced RE is negligible (< 0.1), confirming that the grouping is accurate and the shuffling within communities preserves modularity well. On Enron (Figure 7(d)), $RE \approx 0.3$: this dataset contains 1,363 communities, many with ≤ 10 nodes. The Louvain initialization cannot resolve communities smaller than $k=2400$, so the grouping merges many non-private communities, introducing structural error even in the absence of noise. This suggests that SDPCD’s performance on fine-grained community structures is bounded by the grouping resolution.

Table 3. Average runtime (seconds, $\pm$ std) at $\epsilon_{\text{total}}=1$. Std. omitted if < 1 s.

	<i>Facebook</i>	<i>Government</i>	<i>AstroPh</i>	<i>Enron</i>
RABV	38 ± 6	94 ± 14	830 ± 23	3566 ± 162
LouvainDP	3	10	87 ± 16	1588 ± 23
LDPGen	12	31 ± 4	466 ± 10	1947 ± 45
LFGDPR	129 ± 17	318 ± 36	2371 ± 89	9695 ± 202
SDPCD	12	34 ± 2	358 ± 11	1742 ± 24

**Fig. 7.** RE of modularity for SDPCD and ablations: SDPCD (w/o g) shuffles all Stage-2 edges as one global list (no grouping); SDPCD (w/o n) applies grouping and shuffling to the original unperturbed graph (upper-bound reference, no privacy).

6.4 Parameter Study

Effect of k (Figure 8(a)). Facebook and Government exhibit a clear U-shaped curve with optima at $k=300$ and $k=600$ respectively. For k below the optimum, constraint (4) forces ϵ_{s2} to be small, increasing Stage-2 perturbation noise. For k above the optimum, merging enforces large groups that cross the boundaries of the non-private communities, degrading utility regardless of noise level. AstroPh and Enron exhibit a monotone decrease in RE up to $k=1000$ (the maximum tested), suggesting that their optimal k exceeds 1000.

Effect of ϵ_{total} on optimal k (Figure 8(b)). At low ϵ_{total} (e.g., 0.2), a larger k is needed to reduce the required ϵ_{s2} (and thus Stage-2 noise). At high ϵ_{total} (e.g., 1.0), the noise is modest even for small k , so a smaller k is preferred to preserve community boundaries. This confirms the trade-off: k must balance amplification (which benefits from large k) against structural fidelity (which benefits from small k).

7 Conclusion

We presented SDPCD, a framework that applies privacy amplification by shuffling to differentially private community detection. By separating community-aware grouping from a fresh shuffled release, SDPCD achieves an $(\epsilon_{s1} + \epsilon_{\text{amp}}, \delta)$ -

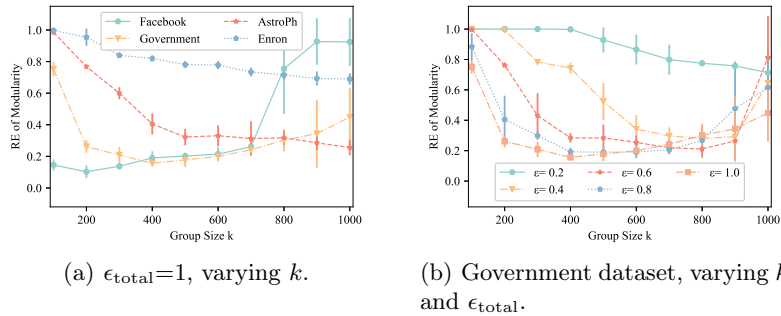


Fig. 8. RE of modularity vs. minimum group size k . Each dataset has an optimal k that balances amplification strength (favors large k) against structural fidelity (favors small k).

Edge DP guarantee against the downstream analyst while consistently improving utility over existing CDP and LDP baselines.

Limitations and future work. The equal budget split $\epsilon_{s1} = \epsilon_{\text{amp}} = \epsilon_{\text{total}}/2$ is a simple heuristic; an adaptive split tuned to the graph’s degree distribution could improve utility. The minimum group size k is chosen by grid search; a principled adaptive method would be preferable. SDPCD assumes non-overlapping communities; extending to overlapping structures is an open problem. Finally, distributing the shuffling functionality across multiple semi-trusted parties would further reduce trust requirements.

Bibliography

- [1] Abouei, A.M., Canonne, C.L.: Differential privacy via group shuffling. In: NeurIPS Workshop on Privacy in Machine Learning. pp. 1–9. arXiv, online (2021)
- [2] Balle, B., Bell, J., Gascón, A., Nissim, K.: The privacy blanket of the shuffle model. In: International Cryptology Conference. pp. 638–667. Springer, Santa Barbara, CA, USA (2019)
- [3] Bassily, R., Nissim, K., Stemmer, U., Guha Thakurta, A.: Practical locally private heavy hitters. *Advances in Neural Information Processing Systems* **30**, 1–9 (2017)
- [4] Bedi, P., Sharma, C.: Community detection in social networks. *Data Mining and Knowledge Discovery* **6**(3), 115–135 (2016)
- [5] Bittau, A., Erlingsson, Ú., Maniatis, P., Mironov, I., Raghunathan, A., Lie, D., Rudominer, M., Kode, U., Tinnes, J., Seefeld, B.: Prochlo: Strong privacy for analytics in the crowd. In: Symposium on operating systems principles. pp. 441–459. ACM, Shanghai, China (2017)
- [6] Blondel, V.D., Guillaume, J.L., Lambiotte, R., Lefebvre, E.: Fast unfolding of communities in large networks. *Journal of statistical mechanics: theory and experiment* **2008**(10), P10008 (2008)

- [7] Chen, R., Fung, B.C., Yu, P.S., Desai, B.C.: Correlated network data publication via differential privacy. *The VLDB Journal* **23**, 653–676 (2014)
- [8] Cheu, A., Smith, A., Ullman, J., Zeber, D., Zhilyaev, M.: Distributed differential privacy via shuffling. In: *International Conference on the Theory and Applications of Cryptographic Techniques*. pp. 375–403. Springer, Darmstadt, Germany (2019)
- [9] Duch, J., Arenas, A.: Community detection in complex networks using extremal optimization. *Physical review E* **72**(2), 027104 (2005)
- [10] Dwork, C.: Differential privacy. In: *International colloquium on automata, languages, and programming*. pp. 1–12. Springer (2006)
- [11] Dwork, C., McSherry, F., Nissim, K., Smith, A.: Calibrating noise to sensitivity in private data analysis. In: *Theory of Cryptography Conference*. pp. 265–284. Springer, New York, NY, USA (2006)
- [12] Erlingsson, Ú., Feldman, V., Mironov, I., Raghunathan, A., Talwar, K., Thakurta, A.: Amplification by shuffling: From local to central differential privacy via anonymity. In: *ACM-SIAM Symposium on Discrete Algorithms*. pp. 2468–2479. ACM, California, USA (2019)
- [13] Feldman, V., McMillan, A., Talwar, K.: Hiding among the clones: A simple and nearly optimal analysis of privacy amplification by shuffling. In: *Annual Symposium on Foundations of Computer Science (FOCS)*. pp. 954–964. IEEE, Colorado, USA (2022)
- [14] Fortunato, S.: Community detection in graphs. *Physics reports* **486**(3-5), 75–174 (2010)
- [15] Imola, J., Murakami, T., Chaudhuri, K.: Differentially private triangle and 4-cycle counting in the shuffle model. In: *ACM SIGSAC Conference on Computer and Communications Security*. pp. 1505–1519. ACM, Los Angeles, USA (2022)
- [16] Jiang, L., Yan, Y., Tian, Z., Xiong, Z., Han, Q.: Personalized sampling graph collection with local differential privacy for link prediction. *Europe PMC* pp. 1–26 (2022)
- [17] Johnson, S.C.: Hierarchical clustering schemes. *Psychometrika* **32**(3), 241–254 (1967)
- [18] Ju, X., Zhang, X., Cheung, W.K.: Generating synthetic graphs for large sensitive and correlated social networks. In: *International Conference on Data Engineering Workshops (ICDEW)*. pp. 286–293. IEEE, Macao, China (2019)
- [19] Kasiviswanathan, S.P., Lee, H.K., Nissim, K., Raskhodnikova, S., Smith, A.: What can we learn privately? *SIAM Journal on Computing* **40**(3), 793–826 (2011)
- [20] Leskovec, J., Krevl, A.: Snap datasets: Stanford large network dataset collection (2014)
- [21] Li, Y., Purcell, M., Rakotoarivelo, T., Smith, D., Ranbaduge, T., Ng, K.S.: Private graph data release: A survey. *ACM Computing Surveys* **55**(11), 1–39 (2021)
- [22] McSherry, F.D.: Privacy integrated queries: an extensible platform for privacy-preserving data analysis. In: *ACM International Conference on*

- Management of data (SIGMOD). pp. 19–30. ACM, Providence, Rhode Island, USA (2009)
- [23] Newman, M.E.: Modularity and community structure in networks. *Sciences* **103**(23), 8577–8582 (2006)
- [24] Nguyen, H.H., Imine, A., Rusinowitch, M.: Differentially private publication of social graphs at linear cost. In: *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*. pp. 596–599. ACM, Paris, France (2015)
- [25] Nguyen, H.H., Imine, A., Rusinowitch, M.: Detecting communities under differential privacy. In: *ACM on Workshop on Privacy in the Electronic Society*. pp. 83–93. ACM, Vienna, Austria (2016)
- [26] Qin, Z., Yu, T., Yang, Y., Khalil, I., Xiao, X., Ren, K.: Generating synthetic decentralized social graphs with local differential privacy. In: *ACM SIGSAC Conference on Computer and Communications Security*. pp. 425–438. ACM, Dallas, USA (2017)
- [27] Vinh, N.X., Epps, J., Bailey, J.: Information theoretic measures for clusterings comparison: is a correction for chance necessary? In: *International conference on machine learning (ICML)*. pp. 1073–1080. ACM, Montreal, Canada (2009)
- [28] Wang, T., Ding, B., Xu, M., Huang, Z., Hong, C., Zhou, J., Li, N., Jha, S.: Improving utility and security of the shuffler-based differential privacy. *VLDB Endowment* **13**(13), 3545–3558 (2019)
- [29] Warner, S.L.: Randomized response: A survey technique for eliminating evasive answer bias. *Journal of the American Statistical Association* **60**(309), 63–69 (1965)
- [30] Xiao, Q., Chen, R., Tan, K.L.: Differentially private network data release via structural inference. In: *ACM International conference on Knowledge discovery and data mining (KDD)*. pp. 911–920. ACM, New York, NY, USA (2014)
- [31] Ye, Q., Hu, H., Au, M.H., Meng, X., Xiao, X.: Lf-gdpr: A framework for estimating graph metrics with local differential privacy. *IEEE Transactions on Knowledge and Data Engineering* **34**(10), 4905–4920 (2020)
- [32] Zhang, Z., et al.: Ldpcd: a novel method for locally differentially private community detection. *Computational Intelligence and Neuroscience* **2022**, 1–18 (2022)

Harmless Opening for Continuous-Valued Local Differential Privacy

Hiroaki Kikuchi¹[0000-0002-0903-8430] and Takao Murakami²

¹ School of Interdisciplinary Mathematical Sciences, Meiji University.
4-21-1 Nakano, Tokyo 164-8525, Japan. kikn@meiji.ac.jp

² Institute of Statistical Mathematics, AIST, RIKEN
tmura@ism.ac.jp

Abstract. Local Differential Privacy (LDP) enables the estimation of statistics of personal data without violating the privacy of individual users. However, poisoning attacks, in which multiple colluding malicious users submit manipulated values, pose a serious threat to the reliability of LDP protocols. To address this, Song et al. proposed “Harmless Opening”, which employs Pedersen Commitments to commit input vectors and allows the server to select values, thereby preventing malicious malicious perturbation. Their approach assumes discrete LDP mechanisms such as GRR and OUE, but extending it to continuous values is not straightforward. We investigate the applicability of Harmless Opening to continuous-valued LDP mechanisms by constructing three protocols: GRR-HO, SR-HO, and PM-HO. We evaluate the proposed scheme in terms of estimation accuracy, robustness against poisoning, and communication cost.

Keywords: Local differential privacy, poisoning attack, commitment

1 Introduction

Privacy Enhancing Technologies (PETs) have become increasingly important for collecting and analyzing personal data while preserving user privacy. Among them, Local Differential Privacy (LDP) enables users to perturb their data locally before transmission, providing strong privacy guarantees without requiring a trusted data collector. LDP has been deployed in large-scale systems by companies such as Google and Apple for collecting usage statistics and telemetry data.

Despite its strong privacy guarantees, LDP remains vulnerable to poisoning attacks. In a poisoning attack, malicious users intentionally submit manipulated reports in order to distort the estimated statistics. Several studies have investigated this problem. Cao et al. [8] analyzed poisoning attacks against frequency estimation protocols and proposed mechanisms that reduce the impact of malicious reports. Wu et al. [9] studied poisoning attacks against key-value LDP protocols and identified conditions under which attacks become ineffective. Li et

Table 1: Comparison of poisoning defenses in LDP.

Work	Numerical Data	Harmless Opening	Robustness
Cao et al. [8]	No	No	–
Wu et al. [9]	No	No	Analysis only
Li et al. [10]	Yes	No	Detection-based
Song et al. [1]	No	Yes	Robust
This work	Yes	Yes	Robust / Mean Robustness (SR)

al. [10] proposed a detection-based approach that identifies suspicious users by comparing estimates obtained from multiple subsets of reports.

More recently, Song et al. [1] proposed *Harmless Opening* (HO), an efficient cryptographic framework that prevents users from bypassing the prescribed perturbation protocol. Instead of detecting malicious users after data collection, HO requires users to commit to candidate perturbation outputs and later reveal part of the committed information. As a result, invalid perturbations can be rejected without revealing the user’s private value. Song et al. demonstrated that HO provides robustness against poisoning attacks for discrete-valued LDP mechanisms such as GRR and OUE.

However, many practical datasets contain numerical attributes, including age, income, location, and sensor measurements. For such data, continuous-valued LDP mechanisms such as Stochastic Rounding (SR) [3] and the Piecewise Mechanism (PM) [4] are commonly used because they provide substantially higher estimation accuracy than discretized frequency-estimation protocols. The applicability of Harmless Opening to these mechanisms has not been investigated.

This paper studies how Harmless Opening can be applied to continuous-valued LDP mechanisms. We consider three approaches: GRR-HO, which applies HO to discretized numerical data; SR-HO, which combines HO with Stochastic Rounding; and PM-HO, which applies HO to the Piecewise Mechanism. Our goal is not to introduce a new cryptographic primitive, but to investigate the practical and theoretical properties of Harmless Opening when applied to numerical data.

The contributions of this paper are summarized as follows.

- We investigate the applicability of Harmless Opening to continuous-valued LDP mechanisms.
- We construct and analyze three variants, GRR-HO, SR-HO, and PM-HO, and discuss their privacy and poisoning-resistance properties.
- We experimentally evaluate their estimation accuracy, observed attack gain, and computational cost using the UCI Adult dataset.
- We show that SR-HO satisfies mean robustness for numerical attributes, whereas PM-HO experimentally exhibits similar behavior and its theoretical robustness is left for future work.

Table 1 summarizes the position of this work with respect to previous studies.

2 Preliminaries

2.1 Local Differential Privacy

A randomized algorithm $\Psi : D \rightarrow \hat{D}$ satisfies ϵ -local differential privacy (ϵ -LDP) if and only if, for any pair of inputs $x_1, x_2 \in D$ and any output $T \in \hat{D}$,

$$\Pr[\Psi(x_1) = T] \leq e^\epsilon \Pr[\Psi(x_2) = T]. \quad (1)$$

LDP ensures that the output distribution generated from an individual's data does not significantly depend on the actual input value. Consequently, an adversary observing the perturbed output cannot reliably distinguish between any two possible inputs.

In this paper, we consider three representative LDP mechanisms: Generalized Randomized Response (GRR), Stochastic Rounding (SR), and the Piecewise Mechanism (PM).

Generalized Randomized Response Let the input be a categorical value $x \in D$, where $d = |D|$ denotes the domain size. GRR perturbs x according to

$$\Pr[\Psi_{\text{GRR}}(x) = y] = \begin{cases} p = \frac{e^\epsilon}{e^\epsilon + d - 1}, & y = x, \\ q = \frac{1}{e^\epsilon + d - 1}, & y \neq x. \end{cases} \quad (2)$$

Given n reports, the frequency of value x can be estimated by

$$\hat{f}(x) = \frac{c(x) - nq}{p - q},$$

where $c(x)$ denotes the number of perturbed reports equal to x .

Stochastic Rounding SR is designed for numerical data. Let the input be a continuous value $x \in [a, b]$. We first normalize it into $[-1, 1]$ by

$$\tilde{x} = -1 + \frac{2(x - a)}{b - a}.$$

Let $p = \frac{e^\epsilon}{1 + e^\epsilon}$, and $q = 1 - p$. The normalized value $\tilde{x}$ is converted into a binary output $x' \in \{-1, 1\}$ according to

$$\Pr[\Psi_{\text{SR}}(\tilde{x}) = x'] = \begin{cases} q + \frac{(p - q)(1 - \tilde{x})}{2}, & x' = -1, \\ q + \frac{(p - q)(1 + \tilde{x})}{2}, & x' = 1. \end{cases} \quad (3)$$

Since $E\left[\frac{x'}{p - q}\right] = \tilde{x}$, an unbiased estimate of the original value can be obtained by $\hat{x} = \frac{\frac{x'}{p - q} + 1}{k} + a$, where $k = \frac{2}{b - a}$.

Piecewise Mechanism The Piecewise Mechanism (PM) also handles continuous values by first normalizing inputs into $[-1, 1]$. Let $s = \frac{e^{\epsilon/2} + 1}{e^{\epsilon/2} - 1}$.

For a normalized input $\tilde{x}$, PM generates an output $x' \in [-s, s]$ according to

$$\Pr[\Psi_{\text{PM}}(\tilde{x}) = x'] = \begin{cases} \frac{e^{\epsilon/2}(e^{\epsilon/2} - 1)}{2(e^{\epsilon/2} + 1)} & \text{if } x' \in [\ell(\tilde{x}), r(\tilde{x})], \\ \frac{e^{\epsilon/2} - 1}{2(e^{\epsilon/2} + e^{\epsilon})}, & \text{otherwise,} \end{cases} \quad (4)$$

where $-s \leq \ell(\tilde{x}) < r(\tilde{x}) \leq s$, $\ell(\tilde{x}) = \frac{e^{\epsilon/2}\tilde{x} - 1}{e^{\epsilon/2} - 1}$ and $r(\tilde{x}) = \frac{e^{\epsilon/2}\tilde{x} + 1}{e^{\epsilon/2} - 1}$. PM achieves lower estimation variance than GRR, especially under relatively small privacy budgets.

2.2 Poisoning Attacks

Although LDP protects the privacy of individual users, it does not guarantee the correctness of submitted reports. Malicious users may intentionally manipulate their reports to distort the aggregated statistics.

Cao et al. [8] analyzed poisoning attacks against discrete LDP mechanisms such as GRR, OUE, and OLH, and proposed three representative attack strategies:

- **Random Perturbed-value Attack (RPA)**: The attacker randomly selects an output from the LDP output domain.
- **Random Item Attack (RIA)**: The attacker randomly selects an input value and then honestly applies the perturbation mechanism.
- **Maximum Gain Attack (MGA)**: The attacker chooses outputs that maximize the change in the estimated frequency of a target item.

Wu et al. [9] further investigated poisoning attacks against key-value LDP protocols, including PrivKV, PCKV-GRR, and PCKV-UE.

The literature commonly distinguishes two adversarial models:

1. **Input Poisoning Attack (IPA)**: The attacker manipulates the input value but follows the prescribed perturbation algorithm.
2. **Output Poisoning Attack (OPA)**: The attacker bypasses the perturbation algorithm and directly submits arbitrary outputs.

Since OPA completely ignores the LDP protocol, it is generally more harmful and more difficult to mitigate.

2.3 Harmless Opening

Output Poisoning Attacks (OPA) are possible because a malicious user can bypass the prescribed perturbation mechanism and submit arbitrary outputs. To

Algorithm 1 VGRR

Require: User $\mathcal{U}$ has input $x \in D$. Server $\mathcal{A}$ collects perturbed y .

1. (*Commit*) $\mathcal{U}$ has ℓ -dimensional vector $\boldsymbol{\mu} = (\mu_1, \dots, \mu_\ell)$ that contains ℓ_1 x s and ℓ_2 values in $D \setminus \{x\}$ and sends to $\mathcal{A}$ the corresponding ℓ -dimensional commitment vector $[\eta] = (\eta_1, \dots, \eta_\ell)$, where $\eta_i = C(\mu_i, \tau_i)$ for random τ_i for $i = 1, \dots, \ell$.
2. (*select*) Server $\mathcal{A}$ picks uniformly $j \in \{1, \dots, \ell\}$, sends it to $\mathcal{U}$.
3. (*open*) $\mathcal{U}$ chooses $J = \{j_1, \dots, j_{\ell_2}\}$ such that $j_1 = j$, and ℓ_2 j' satisfy $\mu_{j'} = i$ for all $i = 1, \dots, \ell$, and discloses to $\mathcal{U}$ the commitments $\eta_{j_1}, \dots, \eta_{j_{\ell_2}}$ and the corresponding random values $\tau_1, \dots, \tau_\ell$.
4. (*verify*) $\mathcal{A}$ verifies J to be $j_1 = j$, $\eta_{j_i} = C(\mu_{j_i}, \tau_{j_i})$ for $i = 1, \dots, \ell$, there are ℓ_2 $i \in J$ such that $\mu_i = x'$ for $x' \in D$. Then, she accepts $y = \mu_j$.

address this problem, Song et al. [1] proposed *Harmless Opening* (HO), a framework for constructing verifiable LDP protocols.

HO is based on an interactive perturbation process between a user and the aggregator. Instead of directly generating a perturbed output, the user first prepares a vector of candidate outputs whose distribution follows the target LDP mechanism. The user then commits to all elements of the vector using Pedersen commitments and sends the commitments to the aggregator. The aggregator randomly selects one commitment, which determines the final perturbed output.

To prevent malicious users from constructing arbitrary vectors, HO requires the user to open additional commitments. These openings are carefully chosen so that they reveal no additional information about the user's private value while restricting the set of valid commitment vectors. Song et al. refer to this property as *harmless opening*.

Based on this idea, Song et al. proposed Verifiable Generalized Randomized Response (VGRR), which combines interactive perturbation, Pedersen commitments, and harmless opening. The complete protocol is summarized in Algorithm 1.

The security of VGRR is established by the following three properties. First, VGRR preserves the privacy guarantee of the underlying GRR mechanism and satisfies $\epsilon = \ln(\ell_1/\ell_2)$ -LDP. Second, the additional information revealed during harmless opening does not provide the aggregator with extra knowledge about the user's private input beyond the perturbed output. Finally, VGRR is robust against poisoning attacks in the sense that the maximum score achievable by any adversarial strategy is reduced to that of an Input Poisoning Attack (IPA), i.e.,

$$Pr[S_{\max} = S_{\text{in}}] > 1 - \text{negl}(\kappa),$$

where S_{in} denotes the maximum score achievable by an IPA, $S_{\max}$ denotes the maximum score achievable against the verifiable protocol, and κ is the security parameter [1].

3 Proposed Methods

3.1 Threat Model

We consider a data collection scenario consisting of n honest users and m malicious users. The malicious users collude with an adversary and attempt to manipulate the estimated mean produced by an LDP protocol.

Let μ and $\hat{\mu}$ denote the true and estimated means, respectively. The adversary aims to maximize or minimize the estimation error by modifying submitted reports. Following Song et al. [1], we consider both Input Poisoning Attacks (IPA) and Output Poisoning Attacks (OPA).

A trivial attack is to continuously submit the maximum value in the domain. However, such behavior can be easily detected. Therefore, we assume that each malicious user initially possesses a value x and participates in the attack only when the value contributes sufficiently to the attack objective. Specifically, we introduce a threshold θ and assume that a malicious user abstains from participation when $x < \theta$.

3.2 GRR-HO

The most straightforward way to apply Harmless Opening to numerical data is to discretize the input domain and execute VGRR on the discretized values.

Let $D = [a, b]$ be the original domain. The domain is partitioned into d disjoint intervals $D_1, \dots, D_d$. For an input value $x \in D_i$, the corresponding discrete symbol i is used as the input of VGRR.

After discretization, the protocol proceeds exactly as Algorithm 1. Therefore, the privacy, indistinguishability, and robustness properties directly follow from those of VGRR.

Theorem 1. *GRR-HO is robust.*

3.3 Robustness for Mean Estimation

The robustness definition of Song et al. [1] is based on frequency estimation, where the attacker's objective is to maximize the estimated frequency of a target item. Since this work considers mean estimation for numerical attributes, we introduce a robustness definition based on the expected estimated mean.

For an attack that maximizes the estimated mean, we define the *attack gain* as

$$G = E[\hat{\mu}_{\text{attack}}] - E[\hat{\mu}_{\text{clean}}], \quad (5)$$

where $\hat{\mu}_{\text{clean}}$ and $\hat{\mu}_{\text{attack}}$ are the estimated means obtained from honest users only, and under a poisoning attack, respectively. Similarly, we say G_{IPA} , G_{OPA} , and G_{max} as the maximum attack gains achievable by IPA, OPA and any adversarial strategy against a verifiable protocol, respectively.

Definition 1. *A verifiable LDP protocol is robust for mean estimation if and only if $G_{\text{max}} = G_{\text{IPA}}$.*

Algorithm 2 SR-HO

Require: User $\mathcal{U}$ has input $x \in [a, b]$. Servr $\mathcal{A}$ collects perturbed y .

1. $\mathcal{U}$ computes ℓ_1, ℓ_2, ℓ such that $\ell_1/\ell = q + (p - q)$, $\ell_2 = \ell - \ell_1$, where $\tilde{x} = -1 + 2(x - a)/(b - a)$.
2. $\mathcal{U}$ lets $x = 1$ if $\tilde{x} > 0$; 0 otherwise and performs Algorithm 1 with $D = \{-1, 1\}$. If $\mathcal{A}$ accepts y , the unbiased estimate $\hat{\mu}$ is $(c(y)/(p - 1) + 1)/k + a$.

Note that preventing output poisoning attacks limits the attacker's influence to that achievable by input poisoning alone.

3.4 SR-HO

We next consider applying Harmless Opening to the Stochastic Rounding (SR) mechanism.

For an input value $x \in [a, b]$, the user first computes $\tilde{x} = -1 + \frac{2(x-a)}{b-a}$, and then determines the probability of outputting +1 according to the SR mechanism.

To construct a verifiable protocol, we represent the binary outputs $\{-1, +1\}$ by a commitment vector and apply the harmless opening procedure of VGRR. The user commits to a vector containing both symbols, and the aggregator verifies that neither symbol is completely eliminated from the committed vector.

Algorithm 2 summarizes the resulting protocol.

Unlike VGRR, the harmless opening procedure does not verify the exact probability distribution required by SR. However, we found that the following interesting property that guarantees the security of SR-HO.

Lemma 1. *For SR-HO, any report accepted by the harmless opening verification procedure satisfies*

$$E\left[\frac{x'}{p - q}\right] \leq 1$$

for mean-maximizing attacks.

Proof. Let $x' \in \{-1, +1\}$ be the output of SR. For an input value $x \in [a, b]$, SR first normalizes x to $\tilde{x} \in [-1, 1]$ and outputs +1 with probability $\Pr[x' = +1] = q + \frac{(p-q)(1+\tilde{x})}{2}$, where $p = \frac{e^\epsilon}{e^\epsilon + 1}$, and $q = 1 - p$. The maximum normalized value is obtained when $x = b$, that is, when $\tilde{x} = 1$. In this case, $\Pr[x' = +1] = p$. In SR-HO, any accepted commitment vector must pass the harmless opening verification. For a mean-maximizing attack, the adversary would like to maximize the number of committed +1 values. However, the verification requires at least ℓ_2 commitments corresponding to -1. Thus, the number of commitments corresponding to +1 is at most $\ell - \ell_2 = \ell_1$. Since the aggregator selects one commitment uniformly at random, $\Pr[x' = +1] \leq \frac{\ell_1}{\ell} = p$. Therefore, $E[x'] = \Pr[x' = +1] - \Pr[x' = -1] = 2\Pr[x' = +1] - 1 \leq 2p - 1 = p - q$. Hence, $E\left[\frac{x'}{p-q}\right] \leq 1$. Since the inverse normalization maps the normalized value 1 to the

endpoint b , no accepted OPA can yield an expected reconstructed value larger than b .

Theorem 2. *SR-HO is robust for mean estimation, i.e., $G_{\max} = G_{\text{IPA}}$.*

Proof. Without loss of generality, we consider an attacker whose objective is to maximize the estimated mean. Under an IPA, the optimal strategy is to submit the maximum input value b . Since the SR estimator is unbiased, $E\left[\frac{x'}{p-q}\right] = 1$, and therefore the maximum expected reconstructed value is b . By Lemma 1, any report accepted by the SR-HO verification also satisfies $E\left[\frac{x'}{p-q}\right] \leq 1$. Hence, no OPA can produce an expected reconstructed value larger than that obtained by the optimal IPA. Therefore, SR-HO is robust for mean estimation.

3.5 PM-HO

The Piecewise Mechanism (PM) perturbs a normalized value $\tilde{x} \in [-1, 1]$ into an output belonging to the interval $[-s, s]$.

To apply Harmless Opening, the interval $[-s, s]$ is discretized into a finite set of values. The user constructs a commitment vector in which the elements inside the interval $[\ell(\tilde{x}), r(\tilde{x})]$ appear with probability proportional to the PM mechanism, while the remaining elements appear with the complementary probability.

The harmless opening procedure verifies that every discretized value appears at least ℓ_2 times in the commitment vector. As in VGRR, an adversary cannot construct a commitment vector that completely removes non-target outputs.

When the discretization granularity is sufficiently large, the protocol approximates the original PM distribution arbitrarily well.

3.6 Discussion

The three proposed protocols provide different levels of protection against poisoning attacks.

GRR-HO directly inherits the robustness guarantee of VGRR after discretization.

PM-HO preserves the essential structure of harmless opening and is expected to provide robustness similar to that of VGRR.

SR-HO is fundamentally different because the harmless opening procedure verifies only the existence of both output symbols and not the exact SR probability distribution.

Unlike GRR-HO, SR-HO guarantees robustness only for mean estimation, rather than Song et al.[1]’s frequency-based robustness.

4 Evaluation

4.1 Experimental Setup

We evaluate the applicability of Harmless Opening (HO) to continuous-valued LDP mechanisms using the **age** attribute of the UCI Adult dataset. The attribute ranges from 17 to 90 years and is treated as numerical data.

The target statistic is the mean age. We evaluate both estimation accuracy and robustness against poisoning attacks. Estimation accuracy is measured by the mean squared error (MSE), while robustness is measured by the observed attack gain, defined as the difference between the estimated mean obtained with and without poisoning attacks.

We investigate the following research questions.

- RQ1: How does discretization affect estimation accuracy?
- RQ2: Does Harmless Opening reduce the impact of poisoning attacks?
- RQ3: Which LDP mechanism provides the best trade-off among accuracy, robustness, and computational cost?

The adversarial model follows Section 3.1. A fraction β of users are malicious. Following Song et al. [1], malicious users participate in an attack only when their original value exceeds a threshold θ . We evaluate both Input Poisoning Attacks (IPA) and Output Poisoning Attacks (OPA).

Unless otherwise stated, each experiment is repeated 20 times and 95% confidence intervals are reported.

4.2 Impact of Discretization

Figure 5 shows the mean value obtained after discretizing the original domain into d intervals. The discretized mean rapidly converges to the true mean, indicating that even coarse discretization introduces only a small approximation error.

However, the estimation error after applying LDP is not necessarily reduced by increasing d . Figure 1 presents the estimated mean under GRR and SR with $\epsilon = 0.1$. While a larger value of d improves the approximation of the original domain, it simultaneously increases the perturbation error introduced by GRR. Consequently, the overall estimation accuracy does not necessarily improve with finer discretization.

This result is consistent with the well-known variance term of GRR, which increases approximately with $\sqrt{d/(n\epsilon)}$. The benefit of finer discretization is therefore offset by the larger perturbation variance.

4.3 Robustness Against Poisoning Attacks

Before evaluating Harmless Opening, we first compare the effect of IPA and OPA on the original GRR mechanism.

Figure 2 shows the estimated mean under poisoning attacks for privacy budgets $\epsilon \in \{0.1, 0.5, \dots, 10\}$. The malicious user ratio is fixed to $\beta = 0.2$.

OPA consistently produces the largest estimation bias, demonstrating the severe vulnerability of conventional LDP protocols. In contrast, IPA only slightly increases the estimated mean even for large privacy budgets. For example, at $\epsilon = 10$, the difference between the poisoned and unpoisoned estimates is approximately 1.03 years under IPA, whereas OPA increases the estimate by more than 10 years.

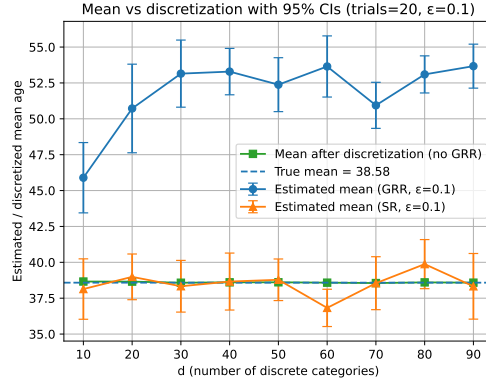


Fig. 1: Estimated mean under GRR and SR with $\epsilon = 0.1$ for different discretization levels d . Although finer discretization improves the representation of the original domain, the perturbation error of GRR increases with d , offsetting the benefit of quantization. Error bars indicate 95% confidence intervals over 20 runs.

These results support the motivation of Harmless Opening. If OPA can be prevented and adversaries are restricted to IPA-like behavior, the impact of poisoning attacks becomes substantially smaller.

4.4 Direct Effect of Harmless Opening

To directly evaluate RQ2, we compare poisoning attacks against GRR with and without Harmless Opening. Figure 3 shows the estimated mean age for four settings: no attack (Base), OPA without HO, selective IPA, and OPA with HO. We use the Adult age attribute with $\beta = 0.2$ and $\theta = 50$, and vary the privacy budget ϵ . Both axes are shown on a logarithmic scale for visibility.

In the OPA setting, malicious users bypass GRR and directly submit the maximum domain value ($b = 90$). In contrast, under OPA+HO, arbitrary outputs are rejected by the harmless opening verification procedure. A malicious user who aims to maximize the estimated mean is therefore constrained to an IPA: the user may manipulate its input as $x = b$, but the submitted value must still be generated through the prescribed GRR mechanism.

The figure shows that HO substantially reduces the effect of OPA, particularly for small privacy budgets. Without HO, directly injecting b produces an extremely large bias because the GRR estimator amplifies manipulated reports when ϵ is small. With HO, the malicious reports must follow the GRR distribution, and the resulting estimate is substantially reduced.

The OPA+HO curve does not coincide with the selective IPA curve. In selective IPA, a malicious user participates only when its original value satisfies $x \geq \theta$ and applies GRR to that original value. In OPA+HO, by contrast, the attacker

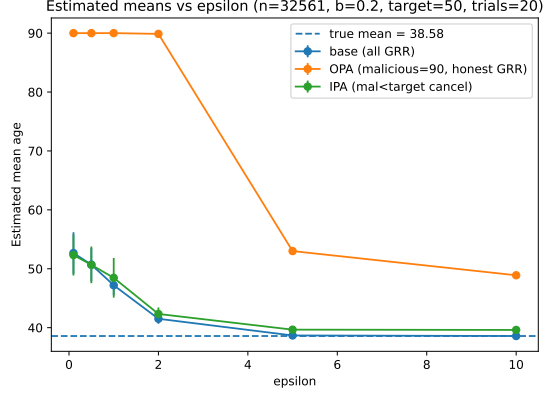


Fig. 2: Estimated mean under GRR in the presence of Input Poisoning Attacks (IPA) and Output Poisoning Attacks (OPA) ($\beta = 0.2$). OPA causes substantially larger estimation bias than IPA, highlighting the importance of preventing malicious users from bypassing the perturbation mechanism.

is allowed to alter its input as $x = b$ before correctly executing GRR. Thus, OPA+HO represents a stronger IPA strategy than the selective IPA considered in our previous experiments.

As ϵ increases, the OPA and OPA+HO estimations become closer. For GRR, the probability of reporting the input value is $p = \frac{e^\epsilon}{e^\epsilon + d - 1}$. Hence, as ϵ increases, p approaches one. Consequently, GRR applied to the modified input $x = b$ increasingly produces $y = b$, making the behavior of OPA+HO closer to that of an unconstrained OPA. This result also indicates that the benefit of HO is most pronounced when randomization is strong, i.e., for relatively small privacy budgets.

Overall, the experiment directly demonstrates the role of Harmless Opening: it prevents an attacker from arbitrarily selecting an LDP output and reduces an OPA to an attack that must obey the underlying perturbation mechanism.

4.5 Accuracy Comparison

Figure 4 compares the estimated mean of GRR, SR, and PM for different privacy budgets and discretization levels.

For all values of d , GRR exhibits increasing estimation error as the domain size grows. In contrast, SR and PM provide stable estimates across a wide range of privacy budgets.

4.6 Observed Attack Gain

Figure 6 shows the observed attack gain under IPA for varying privacy budgets.

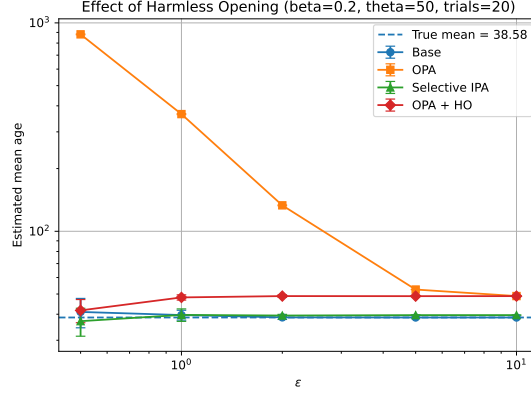


Fig. 3: Effect of Harmless Opening on poisoning attacks against GRR using the Adult age attribute ($\beta = 0.2$, $\theta = 50$)

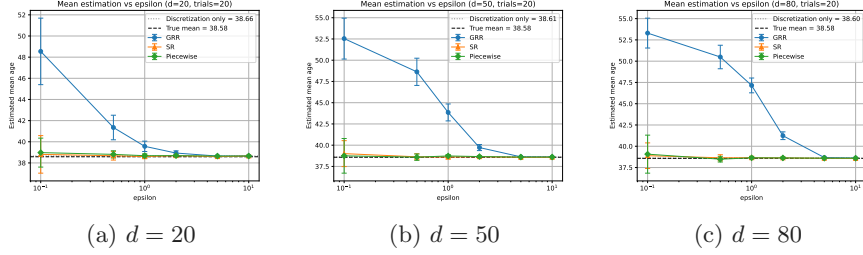


Fig. 4: Estimated mean for GRR, SR, and PM under varying privacy budgets ϵ and discretization levels d . GRR becomes increasingly sensitive to discretization, whereas SR and PM provide stable estimates across a wide range of privacy budgets.

For GRR, the observed attack gain decreases as ϵ increases. In contrast, SR and PM exhibit larger gains and converge toward a stable value around $\epsilon = 5$.

Overall, SR and PM achieve higher estimation accuracy than GRR while maintaining comparable resistance against IPA.

Evaluation on a Different Dataset To examine whether the observations obtained from the Adult dataset depend on a particular numerical attribute or data distribution, we conducted an additional experiment using the Diabetes Prediction Dataset [12]. We selected the HbA1c level as the target numerical attribute, which has a different domain and distribution from the age attribute of the Adult dataset.

We repeated the experiment in Fig. 6 by varying the privacy parameter ϵ from 0.1 to 10. The malicious-user ratio and attack threshold were fixed at $\beta = 0.1$

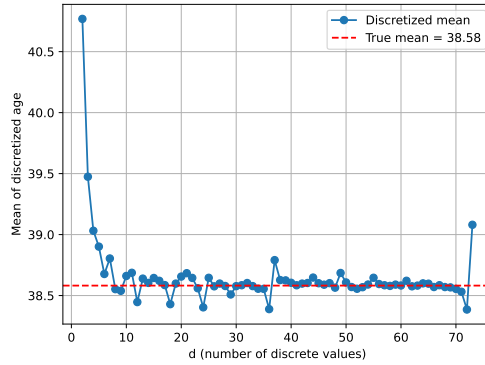


Fig. 5: Mean value after discretizing the age attribute into d bins. The discretized mean rapidly converges to the true mean, indicating that discretization itself introduces only a small approximation error.

and $\theta = 6.0$, respectively. Figure 7 shows the observed attack gain of GRR, SR, and PM for $d = 20, 50$, and 80 .

The detailed behavior differs from that observed for the Adult age attribute, reflecting the different distribution of the HbA1c values. Nevertheless, the main relationship among the three mechanisms is preserved. In particular, for small ϵ , GRR exhibits a substantially larger observed attack gain than SR and PM, and the difference becomes more pronounced as d increases. In contrast, SR and PM show similar behavior and are relatively insensitive to d .

As ϵ increases, the observed attack gains of the three mechanisms converge to similar values. This trend is also consistent with the Adult experiment: with a larger privacy budget, the effect of the perturbation mechanism decreases, and the observed gain is increasingly determined by the IPA strategy itself.

These results suggest that the absolute attack gain and its dependence on ϵ are affected by the underlying data distribution, whereas the main relative behavior of GRR, SR, and PM observed in the Adult dataset is maintained on a different dataset and numerical attribute.

4.7 Computational Cost

Figure 8 shows the execution time of VGRR using Pedersen commitments with 128-, 256-, and 512-bit moduli. The experiments were conducted on a MacBook Pro equipped with a Core i9 processor and 32GB memory.

The computational cost increases approximately linearly with the number of commitments. Since the number of commitments depends on the parameter ℓ , the communication and computation costs of Harmless Opening vary considerably among GRR-HO, SR-HO, and PM-HO.

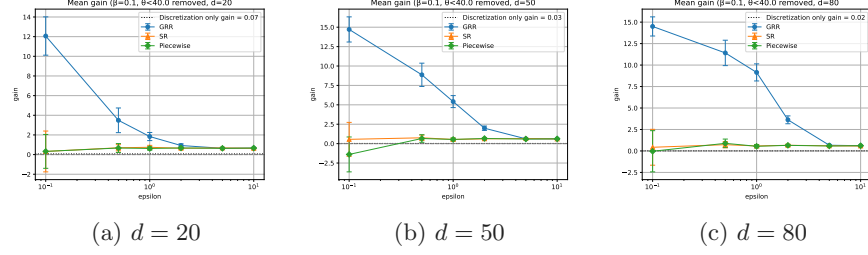


Fig. 6: Observed attack gain under Input Poisoning Attacks (IPA) for varying privacy budgets ϵ and discretization levels d . The malicious user ratio is fixed to $\beta = 0.1$ and the participation threshold is set to $\theta = 40$. GRR exhibits decreasing attack gain as ϵ increases, whereas SR and PM converge toward a stable gain. The results illustrate the trade-off between estimation accuracy and resistance to poisoning attacks for different LDP mechanisms.

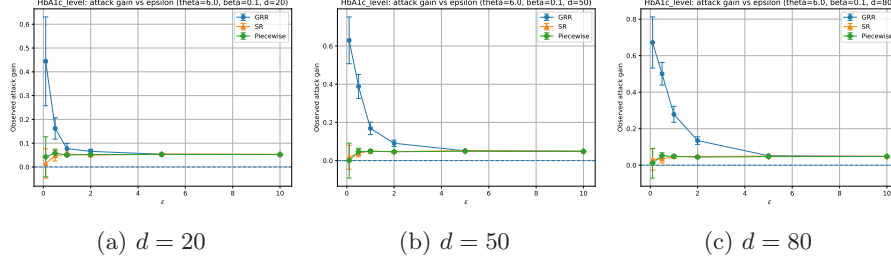


Fig. 7: Observed attack gain for the HbA1c attribute of the Diabetes Prediction Dataset with varying privacy budget ϵ ($\beta = 0.1$, $\theta = 6.0$).

Song et al. [1] reported that VGRR can be executed within approximately 0.05s with negligible communication overhead. Our implementation exhibits similar scalability, although additional optimization is possible.

4.8 Discussion

The experimental results demonstrate that the applicability of Harmless Opening to continuous-valued LDP mechanisms depends strongly on the underlying perturbation mechanism.

GRR-HO directly inherits the robustness guarantee of VGRR. However, discretizing numerical data inevitably enlarges the domain size, resulting in increased perturbation variance and reduced estimation accuracy.

SR-HO provides a different trade-off. Although the harmless opening procedure does not verify the exact SR probability distribution, the proposed robustness definition for mean estimation shows that an adversary cannot increase the expected reconstructed value beyond that achieved by the optimal Input Poisoning Attack. Consequently, SR-HO satisfies *mean robustness* while requiring only

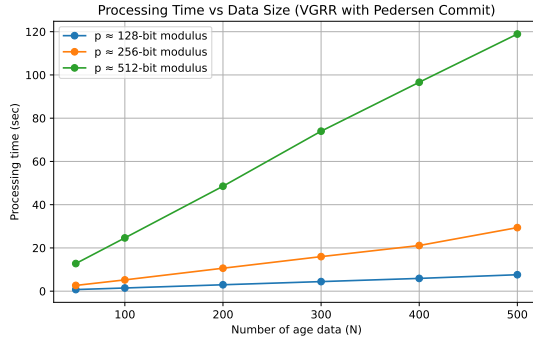


Fig. 8: Execution time of GRR-HO using Pedersen commitments with 128-, 256-, and 512-bit moduli. The experiment was conducted with $\epsilon = \ln 2$, $d = 73$, and $\ell = 200$. The computational cost increases with the commitment size, highlighting the practical overhead introduced by Harmless Opening.

Table 2: Comparison of Harmless Opening applied to different LDP mechanisms

Property	GRR-HO	SR-HO	PM-HO
Target statistic	Frequency	Mean	Mean
Frequency robustness	✓	—	—
Mean robustness	—	✓	Open
Privacy guarantee	ϵ -LDP	ϵ -LDP	ϵ -LDP
Estimation accuracy	Low	High	High
Observed attack gain	Low	Low	Low
Computational cost	High	Low	High

two output symbols, leading to the smallest communication and computational overhead among the three mechanisms.

PM-HO preserves the original perturbation interval more faithfully than GRR-HO and achieves estimation accuracy comparable to SR-HO. However, a formal proof of mean robustness for PM-HO remains an open problem because the harmless opening procedure is constructed by discretizing the continuous output interval.

Table 2 summarizes the characteristics of the three approaches. GRR-HO provides robustness for frequency estimation, whereas SR-HO achieves mean robustness for numerical attributes with substantially lower computational cost. PM-HO experimentally exhibits similar behavior to SR-HO, although its theoretical robustness is left for future work.

5 Conclusion

This paper investigated the applicability of Harmless Opening (HO) to continuous-valued Local Differential Privacy (LDP) mechanisms. While HO was originally

proposed for discrete mechanisms such as GRR, its applicability to numerical data has not been well understood.

To explore this problem, we considered three approaches: GRR-HO, which applies HO after discretization of the input domain; SR-HO, which combines HO with Stochastic Rounding; and PM-HO, which applies HO to the Piecewise Mechanism through discretization of the output domain. We analyzed their security properties and evaluated their performance using the age attribute of the UCI Adult dataset.

The experimental results showed that discretization plays an important role in both estimation accuracy and poisoning resistance. GRR-HO inherits the robustness guarantee of VGRR but suffers from reduced accuracy as the discretization level increases. PM-HO preserves high estimation accuracy and experimentally exhibits resistance to poisoning attacks, although its formal mean robustness remains an open problem. Among the mechanisms considered, SR-HO offers an attractive practical trade-off between accuracy, efficiency, and robustness for mean estimation. SR-HO provides a formal guarantee of mean robustness.

Overall, the results suggest that Harmless Opening can be extended beyond discrete LDP mechanisms and remains effective in mitigating poisoning attacks for numerical data. Among the mechanisms considered, SR-HO offers an attractive practical trade-off between accuracy, efficiency, and observed resistance to poisoning attacks, despite the absence of a formal robustness guarantee.

A promising direction for future work is the design of a verifiable perturbation protocol that directly enforces the probability distribution of continuous-valued LDP mechanisms. Such a protocol could provide both the accuracy of SR and PM and the robustness guarantee achieved by VGRR.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 23K11110 and JST CREST Grant Number JPMJCR21M1.

References

1. S. Song, L. Xu and L. Zhu, “Efficient Defenses Against Output Poisoning Attacks on Local Differential Privacy,” in *IEEE Transactions on Information Forensics and Security*, vol. 18, pp. 5506–5521, 2023.
2. P. Kairouz, K. Bonawitz, and D. Ramage, “Discrete distribution estimation under local privacy,” in *Proc. 33rd Int. Conf. Mach. Learn.*, vol. 48, Jun. 2016, pp. 2436–2444.
3. J. C. Duchi, M. I. Jordan, and M. J. Wainwright, “Minimax optimal procedures for locally private estimation,” *Journal of the American Statistical Association*, vol. 113, no. 521, pp. 182–201, 2018.
4. N. Wang et al., “Collecting and analyzing multidimensional data with local differential privacy,” in *Proc. IEEE 35th Int. Conf. Data Eng. (ICDE)*, Apr. 2019, pp. 638–649.

5. H. Kikuchi, J. Akiyama, H. Gobioff, and G. Nakamura, “Stochastic voting protocol to protect voters privacy,” in Proc. IEEE Workshop Internet Appl., Jul. 1999, pp. 103–111.
6. F. Kato, Y. Cao, and M. Yoshikawa, “Preventing manipulation attack in local differential privacy using verifiable randomization mechanism,” in Data and Applications Security and Privacy XXXV. Cham, Switzerland: Springer, 2021, pp. 43–60.
7. A. Cheu, A. Smith, and J. Ullman, “Manipulation attacks in local differential privacy,” in Proc. IEEE Symp. Secur. Privacy (SP), May 2021, pp. 883–900.
8. X. Cao, J. Jia, and N. Z. Gong, “Data poisoning attacks to local differential privacy protocols,” in Proc. 30th USENIX Secur. Symp. (USENIX Security), 2021, pp. 947–964.
9. Y. Wu, X. Cao, J. Jia, and N. Z. Gong, “Poisoning attacks to local differential privacy protocols for Key-value data,” in Proc. 31st USENIX Secur. Symp. (USENIX Security), 2022, pp. 519–536.
10. X. Li, N. Li, W. Sun, N. Z. Gong, and H. Li, “Fine-grained poisoning attack to local differential privacy protocols for mean and variance estimation,” in Proc. 32nd USENIX Secur. Symp. (USENIX Security), 2023, pp. 1739–1756.
11. P. Zhao et al., “An Attack-Agnostic Defense Framework Against Manipulation Attacks Under Local Differential Privacy,” in 2025 IEEE Symposium on Security and Privacy (SP), pp. 3858–3876, 2025.
12. Kahn, M., “Diabetes [dataset]”, UCI Machine Learning Repository. <https://doi.org/10.24432/C5T59G>, 1994.

Fast Simulation Algorithms for OLH using Binomial Modeling

Berkay Kemal Balioglu, Alireza Khodaie, and
M. Emre Gursesoy

Department of Computer Engineering, Koç University, Istanbul, Turkey
{bbalioglu23, akhodaie22, emregursesoy}@ku.edu.tr

Abstract. Optimized Local Hashing (OLH) is a widely used hash-based Local Differential Privacy (LDP) protocol, and simulation-based experimentation is the standard approach for evaluating OLH and OLH-based applications in research. However, the existing OLH simulations have $O(nd)$ computational complexity, where n is the user population size and d is the domain size, and can lead to significant execution times as n and d grow. In this paper, we propose two fast simulation algorithms for OLH (2-Binom and 3-Binom) grounded in Binomial modeling. Our key insight is that, for any domain value v , the total number of users whose perturbed reports support v can be decomposed into a sum of two or three Binomial random variables. Using this insight, our algorithms reduce the simulation complexity to $O(n + d)$ without hurting statistical equivalence. In particular, we theoretically prove that both algorithms yield unbiased frequency estimations with variances identical to those of the original OLH simulations. Experiments on real-world datasets confirm that both approaches reduce execution times from several minutes to milliseconds, yielding significant speedups with no change in utility.

Keywords: Local differential privacy · optimized local hashing · privacy-enhancing technologies · privacy protocols.

1 Introduction

The growing adoption of data-driven services has increased concerns around user privacy, making privacy-preserving data collection an increasingly critical challenge. Local differential privacy (LDP) has emerged as a response to this challenge, offering mathematically rigorous guarantees that protect individual users without relying on a trusted server [6,9,15,20]. Optimized Local Hashing (OLH) [15] is among one of the most widely adopted LDP protocols – in OLH, each user encodes their true value by applying a randomly drawn hash function before perturbation. This hash-based design makes OLH particularly attractive for large domains; as a result, it has been used extensively in subsequent works and downstream tasks [7,14,16,17,19], and it has been implemented in several LDP benchmarking platforms [1,8,13].

Evaluating LDP protocols and applications built on top of them requires simulation-based experiments. A researcher wishing to assess a new application,

e.g., one that uses OLH as a building block, typically simulates the entire process on a single machine by iterating through each of the n users one by one, accumulating their perturbed outputs, and finally performing server-side estimation to obtain frequency estimates $\hat{f}_v$ for all values v in domain $\mathcal{D}$. This approach accurately reflects real-world deployments but carries a computational burden that grows with both the number of users n and the domain size d . For OLH specifically, there is an $O(nd)$ time complexity, which can cause simulation times of several minutes on moderately sized datasets.

In this paper, we propose two fast simulation algorithms for OLH (called 2-Binom and 3-Binom) that replace the costly user-by-user simulation with draws from a small number of Binomial random variables. Our key insight is that, for any domain value $v \in \mathcal{D}$, users can be partitioned into two groups based on whether their true value equals v or not. The contribution of the first group to the support of v ($\text{Sup}(v)$) follows a Binomial distribution. For the second group, although each user can contribute to $\text{Sup}(v)$ through two ways (a hash collision between their true value and v , or a perturbation to $H_u(v)$) these two sub-cases can be algebraically combined into one Binomial random variable. This yields our 2-Binom algorithm, which reduces the $O(nd)$ complexity to $O(n+d)$, or even $O(d)$ if per-value counts are precomputed and cached. We further propose an alternative algorithm (3-Binom) that keeps the two sub-cases in the second group separate, modeling the number of colliding users as an intermediate Binomial draw before conditioning the remaining perturbation outcomes on it.

We prove that both 2-Binom and 3-Binom yield unbiased frequency estimates with variance identical to that of standard OLH simulation, and that the two algorithms are statistically equivalent to each other despite their different decomposition structures. We also perform experiments on four real-world datasets across a range of privacy budgets and metrics. Results show that 2-Binom and 3-Binom reduce empirical simulation times from several minutes to milliseconds, with no change in the correctness or utility of the simulation results.

In short, our main contributions are as follows:

- We propose two simulation algorithms (2-Binom and 3-Binom) for OLH, in which the collective behavior of n users is captured by drawing random samples from appropriate Binomial random variables.
- We formally prove that both 2-Binom and 3-Binom algorithms yield unbiased frequency estimates with variances equal to the original OLH simulations.
- We empirically validate 2-Binom and 3-Binom on multiple real-world datasets, demonstrating orders-of-magnitude reduction in execution time while producing estimates that are statistically indistinguishable from those of the existing OLH simulation.

2 Background and Preliminaries

2.1 Data Model and Notation

We denote the population of users by $\mathcal{P}$, and the number of users in the population by $n = |\mathcal{P}|$. Each user $u \in \mathcal{P}$ holds a single private value v_u drawn from a

discrete domain $\mathcal{D}$, and we denote the cardinality of the domain by $d = |\mathcal{D}|$. For any value $v \in \mathcal{D}$, we denote by n_v the number of users whose true value equals v , i.e., $n_v = |\{u \in \mathcal{P} : v_u = v\}|$, and by $f_v = n_v/n$ the true frequency of v in the population. Note that $\sum_{v \in \mathcal{D}} n_v = n$ and $\sum_{v \in \mathcal{D}} f_v = 1$.

2.2 Local Differential Privacy (LDP)

LDP is a widely adopted framework for collecting user data while preserving individual privacy. In a typical LDP setting, a server (data collector) wishes to gather statistics from a population of users without learning any individual user's true value v_u . To protect privacy, each user applies a randomized mechanism ψ to their true value v_u locally on their own device, and transmits the perturbed result to the server. Upon receiving perturbed reports from the entire population, the server applies estimation techniques to recover aggregate statistics.

Definition 1 (ϵ -LDP). A randomized mechanism ψ satisfies ϵ -local differential privacy (ϵ -LDP) if and only if, for any two input values $v_1, v_2 \in \mathcal{D}$:

$$\forall y \in \text{Range}(\psi) : \frac{\Pr[\psi(v_1) = y]}{\Pr[\psi(v_2) = y]} \leq e^\epsilon, \quad (1)$$

where $\text{Range}(\psi)$ denotes the set of all possible outputs of ψ .

Intuitively, ϵ -LDP ensures that an observer of the perturbed output y cannot distinguish between any two true values v_1 and v_2 with a likelihood ratio exceeding e^ϵ . The privacy budget ϵ controls the strength of privacy: smaller ϵ yields stronger privacy protection, while larger values permit higher utility.

2.3 Optimized Local Hashing (OLH)

OLH [15] is a hash-based LDP protocol in which each user encodes their true value by applying a randomly chosen hash function that maps domain elements into an integer range $\{0, 1, \dots, g-1\}$ of size $g \geq 2$. The parameter g controls the trade-off between encoding resolution and perturbation likelihood. The choice of g that minimizes estimation variance was shown to be $g = e^\epsilon + 1$ [15].

Let $\mathcal{H}$ denote a universal family of hash functions, where each $H \in \mathcal{H}$ satisfies $H : \mathcal{D} \rightarrow \{0, 1, \dots, g-1\}$. The OLH protocol proceeds as follows.

Encoding and Perturbation. Each user $u \in \mathcal{P}$ independently and uniformly samples a hash function H_u from $\mathcal{H}$ and computes the hash of their true value: $x_u = H_u(v_u)$. The hashed value x_u is then perturbed by reporting $x'_u \in \{0, 1, \dots, g-1\}$ according to:

$$\Pr[x'_u = i] = \begin{cases} p = \frac{e^\epsilon}{e^\epsilon + g - 1}, & \text{if } i = x_u \\ q = \frac{1}{e^\epsilon + g - 1}, & \text{otherwise} \end{cases} \quad (2)$$

The user sends the pair $\langle H_u, x'_u \rangle$ to the server.

Server-side Estimation. Upon receiving reports from all users, the server estimates the frequency of each value $v \in \mathcal{D}$. For a given v , the server first computes the support of v :

$$\text{Sup}(v) = |\{u \in \mathcal{P} : x'_u = H_u(v)\}| \quad (3)$$

which counts the number of users whose reported hash output is consistent with v under their hash function. The estimated frequency $\hat{f}_v$ is then obtained by:

$$\hat{f}_v = \frac{(e^\varepsilon + g - 1)(g \cdot \text{Sup}(v) - n)}{(e^\varepsilon - 1)(g - 1)n} \quad (4)$$

2.4 Simulating OLH: Existing Approach

To empirically evaluate LDP applications, including those involving OLH, researchers rely on simulation-based experiments, i.e., the encoding, perturbation, and estimation steps of all users and the server are executed on a single machine. The representative pseudocode for simulating OLH is given in Algorithm 1. The simulation begins by initializing the values of parameters g and p , as well as an empty list of reports L , which will store the perturbed $\langle H_u, x'_u \rangle$ pairs of each user. Then, in the user-side encoding and perturbation phase (lines 4-12), the simulation iterates through each user $u_i \in \mathcal{P}$ one by one. For each user, a random seed s_{u_i} is sampled to determine H_{u_i} (this is the default strategy to sample each user's hash function H_{u_i} randomly from $\mathcal{H}$ in a way that conforms with typical hash libraries in Python, e.g., `xxhash`). The user's true value v_{u_i} is hashed to produce x_{u_i} , and the hashed value is perturbed according to Equation 2 to determine x'_{u_i} . The perturbed pair $\langle x'_{u_i}, H_{u_i} \rangle$ is stored in L (line 12). After all users have been simulated, the server-side estimation phase begins (lines 13-18). For each value $v \in \mathcal{D}$ to be estimated, its $\text{Sup}(v)$ is computed by iterating through all stored reports in L and counting those whose perturbed hash output is consistent with v under the reported hash function (lines 16-17). Then, $\text{Sup}(v)$ is fed into the OLH estimator to compute $\hat{f}_v$.

The overall complexity of Algorithm 1 is $O(nd)$. The user-side phase is $O(n)$ since it iterates over all n users, but the server-side phase dominates the total runtime with $O(nd)$ complexity because it needs to compute $\text{Sup}(v)$ for all d values. Computing $\text{Sup}(v)$ requires iterating over the list L , which contains n elements. In practice, this dominates the total runtime because of the many hash computations that need to be performed (line 17 is performed $O(nd)$ times). As n and d grow – for instance, n in the order of millions and d in the order of hundreds – this $O(nd)$ cost causes simulation times to grow. Addressing this scalability problem is the primary motivation of our paper.

3 Proposed Methodology

3.1 Mathematical Intuition

We first analyze how the collective effect of OLH can be modeled using Binomial random variables. For value $v \in \mathcal{D}$, we observe that each user $u_i \in \mathcal{P}$ indepen-

Algorithm 1 Existing OLH Simulation

Input: Domain $\mathcal{D}$, budget ε , values of users in $\mathcal{P}$ ($v_{u_1}, v_{u_2}, \dots, v_{u_n}$)
Output: Estimated frequencies $\hat{f}_v$ for all $v \in \mathcal{D}$

- 1: Compute $g \leftarrow \lfloor e^\varepsilon + 1 \rfloor$, $p \leftarrow \frac{e^\varepsilon}{e^\varepsilon + g - 1}$
- 2: Initialize empty list of reports L
- 3: $\triangleright$ **User-side encoding and perturbation**
- 4: **for** each user $u_i \in \mathcal{P}$ **do**
- 5: Sample a random seed s_{u_i} and initialize H_{u_i} accordingly
- 6: $x_{u_i} \leftarrow H_{u_i}(v_{u_i}) \bmod g$
- 7: $r \leftarrow$ sample a random float in $[0, 1]$
- 8: **if** $r \leq p$ **then**
- 9: $x'_{u_i} \leftarrow x_{u_i}$
- 10: **else**
- 11: $x'_{u_i} \leftarrow$ sample uniformly from $\{0, \dots, g-1\} \setminus \{x_{u_i}\}$
- 12: Store $\langle x'_{u_i}, H_{u_i} \rangle$ in L
- 13: $\triangleright$ **Server-side estimation**
- 14: **for** each value $v \in \mathcal{D}$ **do**
- 15: Initialize $\text{Sup}(v) \leftarrow 0$
- 16: **for** each $\langle x'_{u_i}, H_{u_i} \rangle$ in L **do**
- 17: Increment $\text{Sup}(v)$ by $+1$ iff $x'_{u_i} = H_{u_i}(v) \bmod g$
- 18: Compute $\hat{f}_v \leftarrow \frac{(e^\varepsilon + g - 1)(g \cdot \text{Sup}(v) - n)}{(e^\varepsilon - 1)(g - 1)n}$
- 19: **return** $\hat{f}_v$ for all $v \in \mathcal{D}$

dently contributes either 0 or 1 to $\text{Sup}(v)$, and we partition the n users into two groups based on their true value:

Group 1: Users with true value v (n_v users). For a user u_i with $v_{u_i} = v$, their hashed value is $x_{u_i} = H_{u_i}(v_{u_i}) \bmod g = H_{u_i}(v) \bmod g$. This user contributes $+1$ to $\text{Sup}(v)$ if and only if the perturbed output satisfies $x'_{u_i} = x_{u_i}$, which by the perturbation rule in Eq. 2 occurs with probability p . Therefore, each user in Group 1 contributes an independent Bernoulli(p) random variable to $\text{Sup}(v)$, and summing over all n_v such users yields:

$$X \sim \text{Binomial}(n_v, p) \quad (5)$$

Group 2: Users with true value $\neq v$ ($n - n_v$ users). For a user u_i with $v_{u_i} \neq v$, their hashed value is $x_{u_i} = H_{u_i}(v_{u_i}) \bmod g$. This user contributes $+1$ to $\text{Sup}(v)$ if and only if $x'_{u_i} = H_{u_i}(v) \bmod g$, which can occur via two mutually exclusive sub-cases:

Sub-case A: Hash function H_{u_i} causes a hash collision between v_{u_i} and v , i.e., $H_{u_i}(v_{u_i}) \bmod g = H_{u_i}(v) \bmod g$, and also, the perturbation in Eq. 2 retains this value. By the universality of $\mathcal{H}$, the probability of such a hash collision is $\frac{1}{g}$, and the perturbation retains the value with probability p . The probability of this sub-case is therefore $\frac{p}{g}$.

Sub-case B: There is no hash collision, i.e., $H_{u_i}(v_{u_i}) \bmod g \neq H_{u_i}(v) \bmod g$, but the perturbation in Eq. 2 flips x_{u_i} to specifically become $x'_{u_i} = H_{u_i}(v) \bmod g$

g . The probability of no hash collision is $\frac{g-1}{g}$, and the perturbation outputs a specific other value with probability q . The probability of this sub-case is therefore $\frac{(g-1)q}{g}$.

Combining the two sub-cases and substituting $p = \frac{e^\varepsilon}{e^\varepsilon + g - 1}$ and $q = \frac{1}{e^\varepsilon + g - 1}$, the total probability that a user in Group 2 contributes +1 to $\text{Sup}(v)$ is:

$$\frac{p}{g} + \frac{(g-1)q}{g} = \frac{p + (g-1)q}{g} = \frac{\frac{e^\varepsilon}{e^\varepsilon + g - 1} + \frac{g-1}{e^\varepsilon + g - 1}}{g} = \frac{e^\varepsilon + g - 1}{g(e^\varepsilon + g - 1)} = \frac{1}{g} \quad (6)$$

The two sub-cases therefore combine into a single Bernoulli($\frac{1}{g}$) per user, and summing over all $n - n_v$ users in Group 2 yields:

$$Y \sim \text{Binomial}\left(n - n_v, \frac{1}{g}\right) \quad (7)$$

Finally, since X and Y are independent, combining Groups 1 and 2, the total support for any value $v \in \mathcal{D}$ can be expressed as:

$$\text{Sup}(v) = X + Y, \text{ where } X \sim \text{Binomial}(n_v, p), Y \sim \text{Binomial}\left(n - n_v, \frac{1}{g}\right) \quad (8)$$

This decomposition is the key insight underlying our proposed methodology. Rather than simulating all n users individually to compute $\text{Sup}(v)$, we only need to draw two Binomial random variables per domain value $v \in \mathcal{D}$. The only inputs required are n , n_v , p , q , and g , all of which are either determined by the dataset or by the protocol parameters. In the next subsection, we show how this insight translates into a fast simulation algorithm.

3.2 Proposed Simulation Algorithm (2-Binom)

We now translate this mathematical insight into a fast simulation algorithm for OLH. Our proposed 2-Binom simulation algorithm is given in Algorithm 2. 2-Binom replaces the user-by-user simulation with draws from two Binomial random variables. It proceeds in three phases: First, it computes g and p . Second, between lines 2-4, it iterates through the user population once to compute the per-value counts n_v for all $v \in \mathcal{D}$ and the total population size n . Third, between lines 5-10, it iterates over each value $v \in \mathcal{D}$ and applies the Binomial approach from Eq. 8. For each v , two independent Binomial random variables are created and samples are drawn from them: $X \sim \text{Binomial}(n_v, p)$ and $Y \sim \text{Binomial}(n - n_v, \frac{1}{g})$. The sum of the drawn samples gives $\text{Sup}(v)$, from which the estimated frequency $\hat{f}_v$ is computed via the estimator on line 10.

The time complexity of 2-Binom is $O(n + d)$. Computing g and p are $O(1)$. Iterating over all n users to compute the per-value counts n_v is $O(n)$. Iterating over all $v \in \mathcal{D}$ to compute $\hat{f}_v$ using the Binomial approach is $O(d)$ because the loop iterates d times, and the computations performed within each iteration are $O(1)$. Therefore, the overall complexity of Algorithm 2 is $O(n + d)$. Note that this is lower than the $O(nd)$ complexity of Algorithm 1.

Algorithm 2 Proposed 2-Binom Simulation Algorithm

Input: Domain $\mathcal{D}$, budget ε , values of users in $\mathcal{P}$ ($v_{u_1}, v_{u_2}, \dots, v_{u_n}$)
Output: Estimated frequencies $\hat{f}_v$ for all $v \in \mathcal{D}$

- 1: Compute $g \leftarrow \lfloor e^\varepsilon + 1 \rfloor$, $p \leftarrow \frac{e^\varepsilon}{e^\varepsilon + g - 1}$
- 2: $\triangleright$ **Find n_v and n from user values**
- 3: **for** each user $u_i \in \mathcal{P}$ **do**
- 4: $n_{v_{u_i}} \leftarrow n_{v_{u_i}} + 1$, $n \leftarrow n + 1$
- 5: $\triangleright$ **Binomial approach**
- 6: **for** each value $v \in \mathcal{D}$ **do**
- 7: $x \leftarrow$ draw a random sample from $\text{Binomial}(n_v, p)$
- 8: $y \leftarrow$ draw a random sample from $\text{Binomial}\left(n - n_v, \frac{1}{g}\right)$
- 9: $\text{Sup}(v) \leftarrow x + y$
- 10: $\hat{f}_v \leftarrow \frac{\text{Sup}(v) - (n/g)}{n \cdot (p - (1/g))}$
- 11: **return** $\hat{f}_v$ for all $v \in \mathcal{D}$

The complexity of Algorithm 2 can be further reduced to $O(d)$ with a simple but practically important observation. In many LDP experiments, the same dataset is used across multiple simulation runs, e.g., when experiments are repeated to achieve statistical significance, or when ε is varied across a range of values on a fixed dataset. In such cases, the per-value counts n_v and the population size n are identical across all runs. Therefore, lines 2-4 of Algorithm 2 need to be executed only once, and its result can be stored and reused. Subsequent simulation runs then only execute the last phase (lines 5-10), which has complexity $O(d)$. This means that after a one-time $O(n)$ pre-processing cost, an arbitrary number of simulation repetitions can each be completed in $O(d)$ time.

3.3 Utility Analysis of 2-Binom

To formally establish that Algorithm 2 produces correct results (i.e., equivalent to existing OLH simulations), we prove two properties: (i) estimations are unbiased, and (ii) estimation variances are identical to that of Algorithm 1.

Theorem 1 (Unbiasedness). *The frequency estimations $\hat{f}_v$ produced by Algorithm 2 are unbiased, i.e., $\mathbb{E}[\hat{f}_v] = f_v$ for all $v \in \mathcal{D}$.*

Theorem 2 (Variance). *The variance of the frequency estimations $\hat{f}_v$ produced by Algorithm 2 is:*

$$\text{Var}[\hat{f}_v] = \frac{\frac{1}{g} \left(1 - \frac{1}{g}\right)}{n \left(p - \frac{1}{g}\right)^2} + \frac{f_v \left(1 - p - \frac{1}{g}\right)}{n \left(p - \frac{1}{g}\right)} \quad (9)$$

Due to the page limit, proofs of the theorems are given here. To verify the correctness of Theorem 2, we substitute the optimal OLH parameters $g = e^\varepsilon + 1$

and $p = \frac{e^\varepsilon}{e^\varepsilon + g - 1}$ into Eq. 9. Under this parameterization, $\frac{1}{g} = \frac{1}{e^\varepsilon + 1}$, and $p - \frac{1}{g} = \frac{1}{2} - \frac{1}{e^\varepsilon + 1} = \frac{e^\varepsilon - 1}{2(e^\varepsilon + 1)}$. Plugging these into Eq. 9 and simplifying yields:

$$\text{Var}[\hat{f}_v] = \frac{4e^\varepsilon}{n(e^\varepsilon - 1)^2} + \frac{f_v}{n} \quad (10)$$

This matches the known variance of the OLH protocol derived in [15], confirming that Algorithm 2 preserves the variance of the original OLH simulation exactly.

3.4 Alternative Approach: 3-Binom

Recall that in Section 3.1, we divided Group 2 into two sub-cases, and then merged them into a single $Y \sim \text{Binomial}\left(n - n_v, \frac{1}{g}\right)$ at the end. In this section, we present an alternative approach that keeps the two sub-cases separate. Thus, instead of two Binomial random variables, we have three Binomial random variables (hence the name 3-Binom). We show that this alternative formulation also produces unbiased frequency estimates and have identical variance to 2-Binom.

Recall that sub-case A consists of users whose true value is not equal to v , but they have a hash collision. Let n_c denote the number of users who have such a hash collision. A key observation is that n_c is not a fixed quantity, but rather, a random variable. Taking into account the collision probability $\frac{1}{g}$, we have:

$n_c \sim \text{Binomial}\left(n - n_v, \frac{1}{g}\right)$. Given the value of n_c , the $n - n_v$ Group 2 users are split into two sub-cases. Sub-case A consists of n_c users who experienced a hash collision; each such user contributes to $\text{Sup}(v)$ if and only if the perturbation keeps their hashed value, which occurs with probability p . Sub-case B consists of the remaining $n - n_v - n_c$ users who did not experience a hash collision; each such user contributes to $\text{Sup}(v)$ if and only if the perturbation flips their output to the specific value $H_{u_i}(v) \bmod g$, which occurs with probability q . Then, in 3-Binom, summing the contributions of each group yields: $\text{Sup}(v) = X + Y_1 + Y_2$, where: $X \sim \text{Binomial}(n_v, p)$ represents Group 1 users (same as in 2-Binom), $Y_1 \sim \text{Binomial}(n_c, p)$ represents sub-case A of Group 2, and $Y_2 \sim \text{Binomial}(n - n_v - n_c, q)$ represents sub-case B of Group 2.

Algorithm 3 shows the simulation algorithm for 3-Binom. It follows the same structure as Algorithm 2; the main difference is between lines 8-11. Here, to simulate the two sub-cases in Group 2, we first draw n_c , then use it to draw samples for y_1 and y_2 , and finally, $\text{Sup}(v)$ is the summation of 3 random samples. The time complexity is $O(n + d)$.

3.5 Utility Analysis of 3-Binom

We perform a utility analysis for 3-Binom which is similar to 2-Binom. In contrast with 2-Binom, the proofs for 3-Binom require careful treatment of the conditional randomness introduced by n_c , and rely on the law of total expectation to handle the two layers of randomness. We provide the following theorems; due to the page limit, the proofs of the theorems are available here.

Algorithm 3 Proposed 3-Binom Simulation Algorithm

Input: Domain $\mathcal{D}$, budget ε , values of users in $\mathcal{P}$ ($v_{u_1}, v_{u_2}, \dots, v_{u_n}$)
Output: Estimated frequencies $\hat{f}_v$ for all $v \in \mathcal{D}$

- 1: Compute $g \leftarrow \lfloor e^\varepsilon + 1 \rfloor$, $p \leftarrow \frac{e^\varepsilon}{e^\varepsilon + g - 1}$, $q \leftarrow \frac{1}{e^\varepsilon + g - 1}$
- 2: $\triangleright$ **Find n_v and n from user values**
- 3: **for** each user $u_i \in \mathcal{P}$ **do**
- 4: $n_{v_{u_i}} \leftarrow n_{v_{u_i}} + 1$, $n \leftarrow n + 1$
- 5: $\triangleright$ **3-Binom approach**
- 6: **for** each value $v \in \mathcal{D}$ **do**
- 7: $x \leftarrow$ draw a random sample from $\text{Binomial}(n_v, p)$
- 8: $n_c \leftarrow$ draw a random sample from $\text{Binomial}(n - n_v, \frac{1}{g})$
- 9: $y_1 \leftarrow$ draw a random sample from $\text{Binomial}(n_c, p)$
- 10: $y_2 \leftarrow$ draw a random sample from $\text{Binomial}(n - n_v - n_c, q)$
- 11: $\text{Sup}(v) \leftarrow x + y_1 + y_2$
- 12: $\hat{f}_v \leftarrow \frac{\text{Sup}(v) - (n/g)}{n \cdot (p - (1/g))}$
- 13: **return** $\hat{f}_v$ for all $v \in \mathcal{D}$

Theorem 3 (Unbiasedness). *The frequency estimates $\hat{f}_v$ produced by Algorithm 3 are unbiased, i.e., $\mathbb{E}[\hat{f}_v] = f_v$ for all $v \in \mathcal{D}$.*

Theorem 4 (Variance). *The variance of the frequency estimates $\hat{f}_v$ produced by Algorithm 3 is identical to that of Algorithm 2, given by:*

$$\text{Var}[\hat{f}_v] = \frac{\frac{1}{g} \left(1 - \frac{1}{g}\right)}{n \left(p - \frac{1}{g}\right)^2} + \frac{f_v \left(1 - p - \frac{1}{g}\right)}{n \left(p - \frac{1}{g}\right)} \quad (11)$$

4 Experimental Evaluation

In this section, we present the experimental evaluation of our Binomial-based simulation methodologies (2-Binom and 3-Binom) by comparing them against existing OLH simulations (Algorithm 1). Our evaluation has two main goals: (i) to verify that the outputs of 2-Binom and 3-Binom are statistically indistinguishable from those of existing OLH simulations, and (ii) to demonstrate that our methods achieve substantial execution time improvements.

4.1 Experiment Setup

All algorithms were implemented in Python. Simulations were performed on an Intel Core i9-14900K CPU with 64 GB RAM, running Ubuntu 22.04.5 LTS. We used four real-world datasets that are frequently used in the LDP literature:

Adult contains individuals' census-related information¹. We used the age attribute as users' values, which range from 17 to 90, yielding a domain of size $d = 74$. The dataset contains $n = 45,222$ records.

¹ <https://archive.ics.uci.edu/dataset/2/adult>

MSNBC consists of browsing activity logs of users who visited `msnbc.com` on September 28, 1999². Each record corresponds to a sequence of page visits for a user, where each visit is represented by a content category (e.g., news, technology, weather, sports). The domain size is $d = 17$ categories. For each user, we consider only the first entry in their sequence (i.e., the first visited category) as their v_u . The population size is $n = 989,818$.

Kosarak is a clickstream dataset derived from a Hungarian online news portal³. Each record represents the sequence of URLs visited by a user. Due to the sparsity of many URLs, we restricted the domain to $d = 128$ most frequently visited URLs. For users with multiple URLs, the URL with the highest frequency in their sequence was selected as their v_u .

BMS-POS comprises market basket transaction data collected from a large electronics retailer, containing $n = 500,000$ transactions and 1,657 distinct items⁴. We applied the same preprocessing procedure as in Kosarak, retaining the $d = 256$ most frequently purchased items.

Simulations were conducted for privacy budget values $\varepsilon \in \{0.5, 1.0, 1.5, 2.0, 2.5, 3.0, 3.5, 4.0\}$. Execution time experiments are repeated 10 times, and the estimation error and variance experiments are repeated 5,000 times to ensure statistical significance.

4.2 Execution Time Experiments

OLH versus 2-Binom and 3-Binom. We evaluate the execution time of OLH versus 2-Binom and 3-Binom across all four datasets and all ε values. For each configuration, we measure the total time over 10 simulation repetitions. The results are reported in Table 1. It can be observed that 2-Binom and 3-Binom achieve execution times multiple orders of magnitude lower than OLH, completing every configuration in under 5 milliseconds, while OLH ranges from seconds to several minutes. For example, on the MSNBC dataset, OLH requires between 54 and 63 seconds, whereas 2-Binom and 3-Binom complete the same task in under 0.5 milliseconds. On larger datasets such as BMS-POS and Kosarak, OLH times exceed 3-4 minutes, whereas both Binomial methods still finish in well under a second.

The results are consistent with the algorithms’ computational complexities. OLH exhibits an $O(nd)$ cost, causing execution time to increase with both the number of users and the domain size. In particular, datasets with large domains, such as BMS-POS and Kosarak, incur substantially higher runtimes than MSNBC despite MSNBC having the largest population. In contrast, both Binomial methods have lower complexity. It should be noted that 3-Binom incurs a small additional time overhead compared to 2-Binom due to the extra Binomial draws per $v \in \mathcal{D}$, but the effect of this is very small. Overall, these results

² <https://archive.ics.uci.edu/dataset/133/msnbc+com+anonymous+web+data>

³ <https://www.philippe-fournier-viger.com/spmf/index.php?link=datasets.php>

⁴ <https://github.com/cpearce/HARM/blob/master/datasets/BMS-POS.csv>

Table 1. Total execution times (in seconds) for 10 repetitions across different ε values.

Dataset	Method	$\varepsilon=0.5$	$\varepsilon=1.0$	$\varepsilon=1.5$	$\varepsilon=2.0$	$\varepsilon=2.5$	$\varepsilon=3.0$	$\varepsilon=3.5$	$\varepsilon=4.0$
Adult	OLH	8.6496	8.3656	8.1026	7.8320	7.7127	7.6063	7.4373	7.3598
	2-Binom	0.0008	0.0009	0.0008	0.0008	0.0008	0.0008	0.0008	0.0008
	3-Binom	0.0012	0.0014	0.0013	0.0013	0.0013	0.0013	0.0012	0.0013
MSNBC	OLH	63.037	60.954	59.021	57.635	56.345	55.832	54.491	54.867
	2-Binom	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
	3-Binom	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004	0.0004
Kosarak	OLH	159.75	151.31	147.21	141.59	138.33	135.33	133.87	133.36
	2-Binom	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015	0.0015
	3-Binom	0.0018	0.0019	0.0019	0.0019	0.0019	0.0019	0.0018	0.0019
BMS-POS	OLH	304.37	289.84	284.95	274.70	265.88	262.16	257.32	252.03
	2-Binom	0.0025	0.0025	0.0029	0.0025	0.0025	0.0027	0.0025	0.0025
	3-Binom	0.0036	0.0036	0.0041	0.0036	0.0036	0.0036	0.0036	0.0036

demonstrate the substantial efficiency advantage of 2-Binom and 3-Binom over existing OLH simulations.

Impact of parallelizing OLH. The previous experiments used the standard single-threaded OLH simulations as the baseline. To understand whether the gap between OLH and our methods can be closed by parallelizing the OLH simulations, we implemented a multi-threaded version of Algorithm 1 in which the user population is partitioned into η disjoint chunks, and each chunk is assigned to a different CPU thread. Each thread independently computes $Sup(v)$ and $\hat{f}_v$ for its population subset, and the results are combined at the end. We report execution times with $\eta \in \{1, 2, 4, 8, 16\}$ and $\varepsilon = 1$ over 10 repetitions in Table 2, where 2-Binom and 3-Binom remain single-threaded.

We observe that parallelization reduces OLH’s execution time, but its benefit diminishes as more threads are added, and the reduction is not sufficient to close the gap between OLH versus 2-Binom and 3-Binom. Even at $\eta = 16$, OLH still requires several seconds to tens of seconds on every dataset, whereas single-threaded 2-Binom and 3-Binom complete in well under a millisecond throughout. Our methods therefore remain several orders of magnitude faster than OLH at every thread count we tested.

Variability of execution times. To assess run-to-run variability and verify the stability of the reported speedups, we measured the standard deviations of execution times across 10 experiment repetitions. The full results are available here due to the page limit; here, we briefly comment on the takeaways. We observe that OLH exhibits low standard deviations across all configurations. This stability is intuitive because each OLH run lasts several seconds, which reduces the impact of transient system effects. In contrast, 2-Binom and 3-Binom exhibit relatively larger standard deviations compared to their means. Our inspection of individual runs showed that this is caused by a small number of isolated slow executions. It should also be noted that microsecond-scale system-related

Table 2. Total execution times (in seconds) for 10 repetitions at $\varepsilon = 1.0$, comparing 2-Binom and 3-Binom against parallel OLH implementations across a varying number of threads η . The 2-Binom and 3-Binom methods are single-threaded.

Dataset	OLH (parallel, η threads)					Binomial	
	$\eta = 1$	$\eta = 2$	$\eta = 4$	$\eta = 8$	$\eta = 16$	2-Binom	3-Binom
Adult	8.5531	4.3803	2.6380	1.7852	1.4543	0.0005	0.0010
MSNBC	62.739	34.907	20.468	14.625	12.581	0.0002	0.0003
Kosarak	155.14	78.902	48.702	32.588	25.403	0.0009	0.0017
BMS-POS	304.15	154.50	91.823	65.495	52.596	0.0018	0.0034

Table 3. Memory consumptions (in MB) of simulations across different ε values.

Dataset	Method	$\varepsilon=0.5$	$\varepsilon=1.0$	$\varepsilon=1.5$	$\varepsilon=2.0$	$\varepsilon=2.5$	$\varepsilon=3.0$	$\varepsilon=3.5$	$\varepsilon=4.0$
Adult	OLH	3.8444	3.8434	3.8434	3.8433	3.8446	3.8435	3.8433	3.8435
	2-Binom	0.0157	0.0151	0.0152	0.0158	0.0152	0.0158	0.0150	0.0151
	3-Binom	0.0152	0.0157	0.0158	0.0157	0.0157	0.0151	0.0153	0.0157
MSNBC	OLH	83.591	83.591	83.593	83.591	83.591	83.593	83.590	83.591
	2-Binom	0.0150	0.0150	0.0150	0.0152	0.0151	0.0151	0.0153	0.0153
	3-Binom	0.0151	0.0150	0.0150	0.0152	0.0152	0.0151	0.0151	0.0153
Kosarak	OLH	42.138	42.140	42.138	42.142	42.138	42.140	42.140	42.138
	2-Binom	0.0161	0.0161	0.0157	0.0157	0.0156	0.0161	0.0161	0.0154
	3-Binom	0.0157	0.0160	0.0160	0.0161	0.0161	0.0162	0.0161	0.0161
BMS-POS	OLH	42.143	42.139	42.143	42.139	42.139	42.139	42.143	42.139
	2-Binom	0.0165	0.0170	0.0171	0.0166	0.0164	0.0170	0.0170	0.0172
	3-Binom	0.0166	0.0172	0.0169	0.0171	0.0170	0.0171	0.0165	0.0166

impacts are more pronounced in 2-Binom and 3-Binom, since their execution times are on the order of microseconds. More importantly, considering that the 2-Binom and 3-Binom standard deviations are on the order of microseconds, the observed standard deviations are far too small to account for the reported gains. This supports the conclusion that the improvements arise from the proposed algorithms rather than from measurement artifacts.

4.3 Memory Consumption Experiments

Beyond execution time, the memory footprint of simulations can also be important. We compare the peak memory of OLH against 2-Binom and 3-Binom across all four datasets and ε values, measured using Python’s `tracemalloc` on a preloaded dataset so that only the simulation phase is captured. The results are given in Table 3. Across all datasets and privacy budgets, both Binomial methods consume several orders of magnitude less memory than OLH: every Binomial configuration completes within tens of kilobytes, whereas OLH ranges from a few megabytes on Adult to tens of megabytes on MSNBC. Importantly, the effect of

n and d on OLH’s memory differs from their effect on its runtime. Algorithm 1 stores the report $\langle x'_u, H_u \rangle$ of every user in L , giving $O(n + d)$ memory which is dominated by n . This is visible in the results: Kosarak and BMS-POS consume a near-identical 42.14 MB memory despite their domains differing by a factor of two ($d = 128$ versus $d = 256$), because both contain $n = 500,000$ users. On the other hand, MSNBC is the most memory-intensive dataset due to its large n .

In contrast, 2-Binom and 3-Binom store only the per-value counts n_v and the support array, both of size d , so their memory is $O(d)$. Consumption grows only modestly from MSNBC ($d = 17$) to BMS-POS ($d = 256$), even though MSNBC’s population is nearly twice as large. Unlike the execution time results, 2-Binom and 3-Binom are indistinguishable in memory, since the extra Binomial draw in 3-Binom does not require much additional storage. All methods are stable across varying ε , confirming that 2-Binom and 3-Binom offer substantial memory benefits compared to OLH, alongside their execution time benefits.

4.4 Comparison of Estimation Error

To verify that the outputs of 2-Binom and 3-Binom are statistically indistinguishable from those of existing OLH simulations, we compare their estimation errors across all datasets and privacy budgets. We measure error using ℓ_1 and ℓ_2 distances between the estimated frequencies and the true frequencies:

$$\ell_1 \text{ distance} = \sum_{v \in \mathcal{D}} |f_v - \hat{f}_v| \quad \ell_2 \text{ distance} = \sqrt{\sum_{v \in \mathcal{D}} (f_v - \hat{f}_v)^2} \quad (12)$$

We report the results in Figure 1. It can be observed that the curves for OLH, 2-Binom, and 3-Binom are nearly indistinguishable across all datasets and privacy budgets, indicating that the proposed simulations accurately reproduce the behavior of OLH. For all methods, both ℓ_1 and ℓ_2 distances decrease as ε increases, reflecting the reduction in perturbation noise and the resulting improvement in estimation accuracy. Overall, the proposed methods achieve the same estimation results as OLH while providing the substantial time savings shown in Table 1.

To further examine the distributional similarity of the three methods, we use kernel density estimation (KDE) to visualize the full distribution of ℓ_1 and ℓ_2 distances across the 5,000 repetitions at $\varepsilon = 2.0$. The resulting density plots are shown in Figure 2, with ℓ_1 distributions in the top row and ℓ_2 distributions in the bottom row. In all plots, the three density curves for OLH, 2-Binom, and 3-Binom overlap closely. This indicates that the distributions of estimation errors produced by OLH, 2-Binom, and 3-Binom are nearly identical. Furthermore, the KDE plots reveal that the distributions are approximately bell-shaped in all datasets, with the three methods sharing the same center and spread. Any visible separation between the curves is minor and plausible due to the randomized nature of OLH. The tightest agreement is observed on MSNBC ($d = 17$), where the small domain causes each method’s per-repetition noise to be highly concentrated, while on BMS-POS and Kosarak the distributions are slightly wider owing to the larger domains.

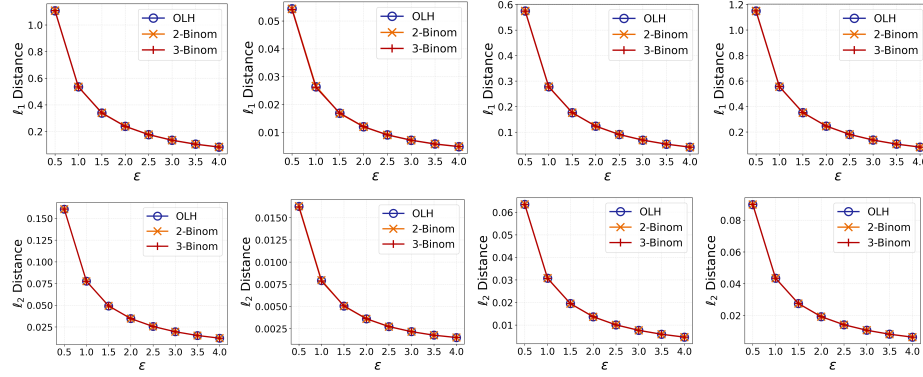


Fig. 1. Average ℓ_1 and ℓ_2 estimation errors of OLH, 2-Binom, and 3-Binom across varying ϵ values. Datasets from left to right: Adult, MSNBC, Kosarak, BMS-POS.

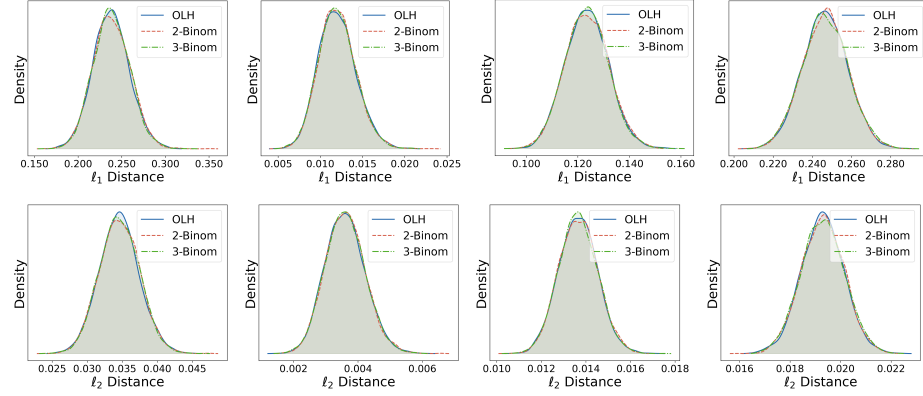


Fig. 2. KDE of ℓ_1 and ℓ_2 distances across 5,000 repetitions at $\epsilon = 2.0$. Datasets from left to right: Adult, MSNBC, Kosarak, BMS-POS.

4.5 Comparison of Empirical Variance

Finally, we verify that OLH, 2-Binom, and 3-Binom agree in terms of per-value variances. We perform 5,000 repetitions for each method at $\epsilon = 2.0$ and compute the empirical variance of $\hat{f}_v$ for each $v \in \mathcal{D}$. The results are shown in Figure 3, where domain values on the x -axis are sorted in descending order of their true frequency f_v , so that high-frequency values appear on the left and low-frequency values on the right.

Across all four datasets, the empirical variance curves of OLH, 2-Binom, and 3-Binom track each other closely and follow the same decreasing trend. This is consistent with our theoretical result in Section 3.3, and the fact that variance is positively correlated with f_v . We note that the shapes of the variance curves may differ across different datasets, which is caused by the frequency distributions of the datasets. On MSNBC, the variance drops sharply from index 0 to index 2 and then flattens into a nearly horizontal plateau, since the dataset has two dominant

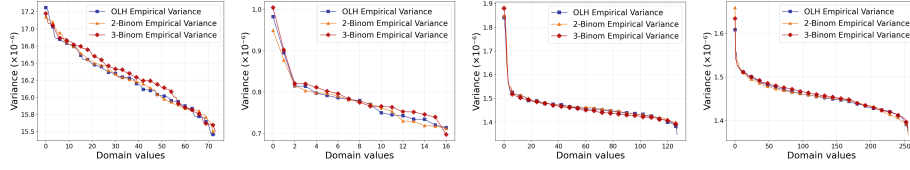


Fig. 3. Empirical variances of OLH, 2-Binom, and 3-Binom at $\varepsilon = 2.0$. Datasets from left to right: Adult, MSNBS, Kosarak, BMS-POS.

categories whose high f_v causes a distinctly elevated variance for those values. On Kosarak and BMS-POS, the initial drop is steep but the curves decay more gradually over the wider domain, reflecting the long-tail frequency distribution of the datasets. On Adult, the decay is more gradual and uniform, consistent with the smoother age distribution in that dataset.

Overall, in all cases, the empirical variance curves of 2-Binom and 3-Binom closely follow those of OLH. This confirms that both methods preserve not only the mean estimation accuracy of OLH but also its per-value variances, providing empirical validation of the theoretical guarantees established in Section 3.3.

5 Related Work

LDP has been an active area of research [6,20], with several frequency estimation protocols proposed and studied in the literature, including Generalized Randomized Response (GRR) [11], RAPPOR [9], Optimized Unary Encoding (OUE), and Optimized Local Hashing (OLH) [15]. Among them, OLH is particularly attractive for large d due to its per-user communication cost and optimized variance, and it has become one of the most widely adopted protocols in the LDP literature. For example, Wang et al. [16,17] used OLH in frequent itemset mining and heavy hitter identification. Cormode et al. [7] proposed methods for hierarchical range query answering via LDP frequency estimators such as OLH. Xu et al. [18] used OLH for analyzing data jointly from multiple services under LDP. Qian et al. [14] used OLH for multi-dimensional value collection under LDP. Xu et al. [19] proposed MLPKV for private key-value data collection using OLH.

There were also studies conducted regarding the fairness and adversarial robustness aspects of LDP protocols, including OLH. Gursoy et al. [10] analyzed protocols, including OLH, from a Bayesian adversarial perspective. Arcolezi et al. [3] demonstrated the risks of collecting multidimensional data under LDP and used OLH as one of the protocols. Cao et al. [5] demonstrated data poisoning attacks on LDP protocols, and used OLH as one of the protocols. Balioglu et al. [4] specifically studied OLH by focusing on the privacy and poisoning-related impacts of hash function choice. Many other works can also be found which utilize OLH; thus, many works would benefit from our contribution of increasing the speed of OLH simulations.

Several simulation and benchmarking platforms for LDP protocols have been developed. Cormode et al. [8] introduced PURE-LDP, a Python benchmarking package implementing OLH, OUE, and other protocols. Arcolezi et al. [1] developed MULTI-FREQ-LDP, a toolkit supporting one-time, multidimensional, and

longitudinal frequency estimation with local hashing. Arcolezi and Gambs [2] built LDP-Auditor, a framework for empirically auditing the privacy loss of LDP protocols through distinguishability attacks across eight frequency estimation protocols. Khodaie et al. [13] introduced a benchmarking platform evaluating LDP protocols and post-processing methods across multiple datasets and utility metrics. The existing OLH implementations in all of these platforms follow the standard $O(nd)$ simulation approach; our proposed method can serve as a drop-in replacement to improve their execution times.

Finally, our work builds on Karatas and Gursoy [12], who proposed Binomial modeling for bitvector-based LDP protocols (RAPPOR and OUE), reducing their simulation complexity from $O(nd)$ to $O(n + d)$. Extending their approach to OLH is non-trivial, as OLH has a different perturbation mechanism: users whose true value differs from v can contribute to $\text{Sup}(v)$ through either a hash collision or a perturbation event, requiring a distinct Binomial decomposition.

6 Conclusion

In this paper, we addressed the $O(nd)$ simulation cost of OLH, a widely used hash-based LDP protocol, by introducing two Binomial-based simulation algorithms: 2-Binom and 3-Binom. Our algorithms replace user-by-user simulation with draws from a small number of Binomial random variables, reducing complexity to $O(n + d)$ and potentially further to $O(d)$ per repetition when counts are precomputed. We formally proved that both algorithms yield unbiased frequency estimates and carry variance identical to that of the original OLH simulation. Our algorithms were also validated empirically using four real-world datasets and a range of ϵ budgets. Results show that 2-Binom and 3-Binom match the estimation results of standard OLH simulation while reducing execution times from several minutes to milliseconds. In future work, we plan to incorporate 2-Binom and 3-Binom into LDP benchmarking platforms and apply the speedup to downstream OLH-based applications.

Acknowledgments

This work was supported by the Scientific and Technological Research Council of Türkiye (TUBITAK) under grant number 123E179 and the BAGEP Outstanding Young Scientist Award. The authors thank TUBITAK and the Science Academy for their support.

References

1. Arcolezi, H.H., Couchot, J.F., Gambs, S., Palamidessi, C., Zolfaghari, M.: Multi-freq-LDPy: Multiple frequency estimation under local differential privacy in python. In: European Symposium on Research in Computer Security (ESORICS), pp. 770–775. Springer Nature Switzerland (2022)
2. Arcolezi, H.H., Gambs, S.: Revealing the true cost of locally differentially private protocols: An auditing perspective. *Proceedings on Privacy Enhancing Technologies* **2024**(4), 123–141 (2024)

3. Arcolezi, H.H., Gambs, S., Couchot, J.F., Palamidessi, C.: On the risks of collecting multidimensional data under local differential privacy. *Proceedings of the VLDB Endowment* **16**(5), 1126–1139 (2023)
4. Balioglu, B.K., Khodaie, A., Gursay, M.E.: Don't hash me like that: Exposing and mitigating hash-induced unfairness in local differential privacy. In: *European Symposium on Research in Computer Security*. pp. 22–42. Springer (2025)
5. Cao, X., Jia, J., Gong, N.Z.: Data poisoning attacks to local differential privacy protocols. In: *30th USENIX Security Symposium*. pp. 947–964 (2021)
6. Cormode, G., Jha, S., Kulkarni, T., Li, N., Srivastava, D., Wang, T.: Privacy at scale: Local differential privacy in practice. In: *Proceedings of the International Conference on Management of Data*. pp. 1655–1658. ACM (2018)
7. Cormode, G., Kulkarni, T., Srivastava, D.: Answering range queries under local differential privacy. *Proceedings of the VLDB Endowment* **12**, 1126–1138 (06 2019)
8. Cormode, G., Maddock, S., Maple, C.: Frequency estimation under local differential privacy. *Proceedings of the VLDB Endowment* **14**(11), 2046–2058 (2021)
9. Erlingsson, Ú., Pihur, V., Korolova, A.: RAPPOR: Randomized aggregatable privacy-preserving ordinal response. In: *Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security (CCS)*. pp. 1054–1067
10. Gursay, M.E., Liu, L., Chow, K.H., Truex, S., Wei, W.: An adversarial approach to protocol analysis and selection in local differential privacy. *IEEE Transactions on Information Forensics and Security* **17**, 1785–1799 (2022)
11. Kairouz, P., Bonawitz, K., Ramage, D.: Discrete distribution estimation under local privacy. In: *International Conference on Machine Learning*. pp. 2436–2444. PMLR (2016)
12. Karatas, Y.C., Gursay, M.E.: Accelerating simulations of bitvector-based LDP protocols via binomial modeling. In: *Proceedings of the 20th International Workshop on Security (IWSEC 2025)*. *Lecture Notes in Computer Science*, vol. 16208, pp. 307–325. Springer (2026)
13. Khodaie, A., Balioglu, B.K., Gursay, M.E.: Post-processing in local differential privacy: An extensive evaluation and benchmark platform. In: *IFIP International Conference on ICT Systems Security and Privacy Protection*. pp. 76–90. Springer (2025)
14. Qian, Q., Ye, Q., Hu, H., Huang, K., Chan, T.T.L., Li, J.: Collaborative sampling for partial multi-dimensional value collection under local differential privacy. *IEEE Transactions on Information Forensics and Security* **18**, 3948–3961 (2023)
15. Wang, T., Blocki, J., Li, N., Jha, S.: Locally differentially private protocols for frequency estimation. In: *Proceedings of the 26th USENIX Security Symposium (USENIX Security 17)*. pp. 729–745. USENIX Association, Vancouver, BC (2017)
16. Wang, T., Li, N., Jha, S.: Locally differentially private frequent itemset mining. *2018 IEEE Symposium on Security and Privacy (SP)* pp. 127–143 (2018)
17. Wang, T., Li, N., Jha, S.: Locally differentially private heavy hitter identification. *IEEE Transactions on Dependable and Secure Computing* **18**(2), 982–993 (2019)
18. Xu, M., Ding, B., Wang, T., Zhou, J.: Collecting and analyzing data jointly from multiple services under local differential privacy. *Proc. VLDB Endow.* **13**(11), 2760–2772 (2020)
19. Xu, X., Fan, Z., Trovati, M., Palmieri, F.: Mlpkv: A local differential multi-layer private key-value data collection scheme for edge computing environments. *IEEE Transactions on Information Forensics and Security* **18**, 1825–1838 (2023)
20. Yang, M., Guo, T., Zhu, T., Tjuawinata, I., Zhao, J., Lam, K.Y.: Local differential privacy and its applications: A comprehensive survey. *Computer Standards & Interfaces* **89**, 103827 (2023)

The Privacy Peak: Non-Monotonic Leakage in Federated Learning

Hamza Sajjad^{1,2}[0000–0002–6954–938X], Xavier Lessage², Gabriele Costa¹[0000–0002–9385–3998], and Phillipe Massonet²

IMT School for Advanced Studies, Lucca, Italy¹
CETIC asbl – Centre d’Excellence en Technologies de l’Information et de la
Communication, Belgium²
hamza.sajjadahmed@imtlucca.it
xavier.lessage@cetic.be gabriele.costa@imtlucca.it
philippe.massonet@cetic.be

Abstract. Federated Learning (FL) enables collaborative model training without direct access to decentralized client data but because of the shared global model it still remains vulnerable to privacy leakage. In this work, we study post-hoc data reconstruction attacks, where an adversary recovers training data using only the final trained global model under heterogeneous (non-IID) data distributions across clients. We propose an anchor-guided reconstruction framework that initializes reconstruction from partially preserved regions of auxiliary images, referred to as anchors, rather than from random noise, and further guides optimization using auxiliary feature priors. Our method relies solely on the global model without accessing the client data, updates or training dynamics. We evaluate the framework on CIFAR-10, a standard 10-class image classification benchmark, across multiple heterogeneity regimes ranging from IID to extreme non-IID settings. Results showed that moderate non-IID settings produced the highest reconstruction fidelity and semantic consistency whereas extreme non-IID settings reduce reconstruction quality primarily because of reduced model quality rather than enhanced privacy. Overall, our results reveal a non-monotonic relationship between data heterogeneity and privacy leakage under the evaluated CIFAR-10 FL setting, showing that federated models retain recoverable training information under strong distribution shifts.

Keywords: Federated Learning · Data Privacy · Model Inversion · Reconstruction Attack

1 Introduction

Federated Learning (FL) is a promising paradigm for collaborative model training that avoids centralizing sensitive user data [26,13]. In cross-device FL, each client trains their own model using private data and sends the update to a central server that combines it into a global model. This decentralized approach is

increasingly adopted in privacy-sensitive domains such as healthcare, mobile applications, and personalized recommendation systems [4,14]. Despite its privacy-preserving motivation, FL does not necessarily provide data privacy on its own. Information regarding the local training data may be embedded within the model updates, which can then be used by adversaries who monitor the training process [22,7]. Privacy risks include membership inference [20], attribute inference, and data reconstruction attacks, challenging the assumption that FL alone ensures privacy [18,19]. It is, therefore, essential to analyze these threats in practical deployment scenarios, especially those involving heterogeneous client datasets.

A key challenge in understanding privacy leakage arises from non-independent and identically distributed (non-IID) client data, which naturally results from differences in user behavior, environments, and device usage [27]. Such heterogeneity affects model convergence, accuracy, and learned representations [28,1], potentially changing the information retained by the global model and its susceptibility to reconstruction attacks. Therefore, evaluating different heterogeneity levels is important for understanding privacy leakage under realistic FL conditions. Existing reconstruction attacks in FL primarily rely on gradient leakage, feature matching, or optimization-based inversion techniques [6,5,10]. These attacks have shown that shared gradients and updates may leak confidential training data. However, such approaches can suffer from unstable optimization and limited semantic consistency [10]. For example, reconstruction from purely random initialization often yields semantically weak results, especially under non-IID data distributions commonly observed in practice [23].

In this work, we focus on a post-hoc reconstruction setting: the attack occurs after federated training, and the adversary has access only to the final global model, not client gradients, updates, or training information. This lets us study how much private information remains embedded in the deployed model independently of the training process. We introduce an anchor-guided reconstruction framework, where an “anchor” is an auxiliary image from the same semantic domain that provides partial visual and feature-space guidance. Rather than starting from random noise, our approach initializes reconstruction using partially preserved anchor regions combined with stochastic noise. By combining internal feature representations, Batch Normalization statistics, perceptual consistency, and anchor-preservation constraints, this strategy guides optimization toward realistic, class-representative reconstructions. We evaluate our method on CIFAR-10, a standard 10-class image classification benchmark, under IID, mild non-IID, strong non-IID, and extreme non-IID settings. Our results show that anchor initialization and feature-bank guidance, which uses auxiliary model representations as semantic priors, primarily improve reconstruction quality, while multi-anchor aggregation improves consistency across anchor locations. Existing work has largely focused on IID or mildly non-IID conditions, leaving reconstruction attacks under stronger heterogeneity insufficiently understood. Our findings show that model representations retain recoverable information even under highly heterogeneous training conditions, highlighting persistent privacy risks. The main contributions of this paper are summarized as follows:

1. We conduct a systematic study of privacy leakage in FL across multiple heterogeneity regimes ranging from IID to extreme non-IID settings, revealing an underexplored non-monotonic relationship between data heterogeneity and reconstruction vulnerability within the evaluated experimental setting.
2. We propose an anchor-guided reconstruction framework for post-hoc global model inversion that combines partial-anchor initialization, auxiliary feature-bank guidance, and multi-objective optimization using only the final trained global model.
3. We demonstrate that reconstruction fidelity peaks under strong non-IID conditions rather than under IID or extreme non-IID settings, challenging the assumption that increasing heterogeneity inherently improves privacy.
4. Through quantitative evaluation, statistical significance testing, ablation studies, and qualitative analysis, we show that federated models retain semantically meaningful information about training data even when adversaries have no access to gradients, local updates, or training dynamics.

The remainder of this paper is organized as follows. Section 2 reviews prior work on FL and privacy leakage. Section 3 defines the adversarial assumptions considered in this work. Section 4 presents the proposed reconstruction framework. Section 5 describes the experimental setup and results. Section 6 discusses the implications of the findings, Section 7 outlines limitations and future work, and Section 8 concludes the paper.

2 Related Work

Federated Learning under Data Heterogeneity. FL enables multiple clients to collaboratively train a shared global model without exchanging raw data [13,17]. Federated Averaging (FedAvg) is the most widely used optimization method due to its simplicity and communication efficiency [21]. However, its performance degrades significantly under non-IID data distributions, which naturally arise from differences in user behavior, device characteristics, and data availability [27]. To simulate such conditions, prior work commonly uses Dirichlet-based partitioning, where the concentration parameter α controls the degree of heterogeneity [11]. To mitigate performance degradation, several optimization techniques have been proposed, including proximal regularization methods such as FedProx [15] and variance reduction approaches based on control variates [12]. While these methods improve convergence and model utility, their privacy implications remain less explored.

Privacy Leakage in Federated Learning. Even though FL does not disclose the raw data, there is a possibility of leakage through model parameters and updates. This is because membership inference attacks and attribute inference attacks have been proposed previously, where the former attack determines whether a given sample is used in the training phase [2], and attribute inference attacks

recover statistical properties of client datasets [16]. The severity of leakage depends on multiple factors, including model architecture, optimization dynamics, and attacker assumptions [22,7]. However, despite increasing efforts towards studying privacy leakage in FL, most existing studies consider a limited level of heterogeneity in their experiments. As a result, the impact of realistic non-IID distributions on privacy leakage remains insufficiently understood.

Reconstruction Attacks. The most serious privacy concerns associated with FL include reconstruction attacks, which seek to recover training data by exploiting gradients, model parameters, or intermediate representations. Initial attempts like Deep Leakage from Gradients (DLG) [29] along with more recent techniques based on gradient matching and optimization-based inversion [8,3] demonstrated that perfect reconstructions could be achieved with minimal information leakage.

More recently, optimization-based model inversion methods have explored post-hoc reconstruction directly from trained models without requiring access to gradients or client updates. For example, DeepInversion [25] synthesizes representative images by optimizing random noise using Batch Normalization statistics and prediction consistency encoded in a pretrained model. While such approaches demonstrate that trained models retain recoverable semantic information, they typically initialize reconstruction from random noise and focus on image synthesis or knowledge distillation rather than analyzing privacy leakage under heterogeneous FL settings.

Later studies extended reconstruction attacks to FL by exploiting correlations between client updates and local data distributions [19,9]. Additional work further explored reconstruction under partial observability and aggregated updates [24].

Limitations in Prior Work. Existing studies of FL privacy leakage exhibit three main limitations. First, most evaluations focus on IID or a single heterogeneity level, limiting understanding under realistic non-IID conditions. Second, reconstruction quality is often measured primarily by optimization success, without sufficient emphasis on semantic consistency and stability. Third, relatively little attention has been given to post-hoc reconstruction attacks that rely solely on the final deployed global model without requiring gradients, client updates, or training dynamics.

However, existing reconstruction methods either assume access to gradients, client updates, intermediate activations, or other training-time information, or focus on generic optimization-based inversion without considering heterogeneous FL settings. Such assumptions are not always realistic once federated training has completed and only the deployed global model remains available. In contrast, we consider a strictly post-hoc threat model in which the adversary has access only to the final trained global model. Rather than proposing reconstruction solely as an end goal, we use it as an empirical tool to investigate how data heterogeneity influences model invertibility and privacy leakage across FL deployments.

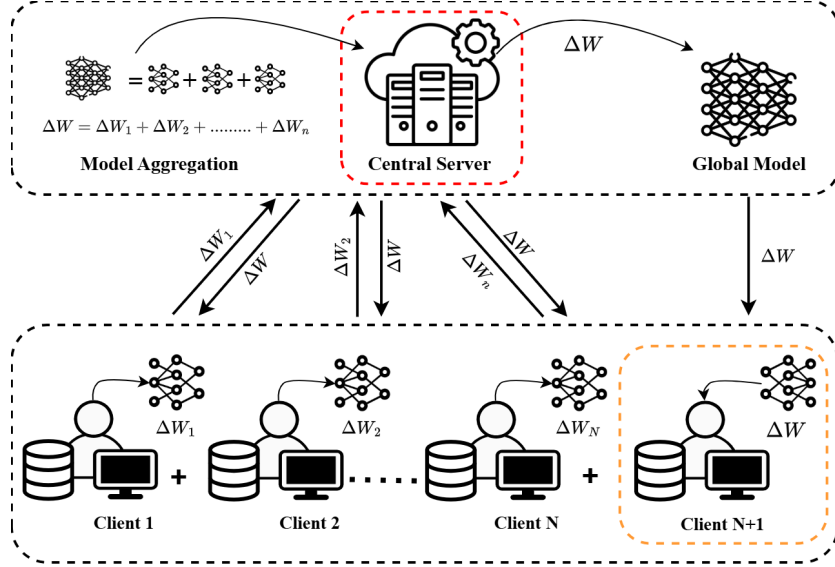


Fig. 1. FL threat model. After training, either the server or a participating client may act as a post-hoc adversary with white-box access to the final global model.

3 Threat Model and Attack Setting

We consider the FL setting illustrated in Figure 1. A central server coordinates training across K distributed clients, where each client k holds a private dataset $\mathcal{D}_k$. Clients perform local optimization and periodically send model updates to the server, which aggregates them using FedAvg. Raw training data never leaves client devices.

3.1 Adversary and Data Assumptions

We consider a post-hoc adversary, either the server or a participating client, with white-box access to the final trained global model f_θ , including all model parameters, Batch Normalization statistics, and intermediate feature representations, as well as the ability to query the model for output logits given arbitrary inputs. The adversary has no access to other clients' private datasets, local gradients or model updates, optimizer states, or any intermediate training-time information. This setup isolates privacy leakage originating from the final global model parameters, rather than from client updates or training dynamics. Client data distributions are either IID or non-IID. We model heterogeneity using Dirichlet partitioning [11], where client class proportions are sampled from a Dirichlet distribution controlled by concentration parameter α ; smaller α produces more skewed client distributions. This assumption is consistent with prior data-assisted model inversion and representation-guided reconstruction literature, where publicly available data from the same semantic domain is used as a

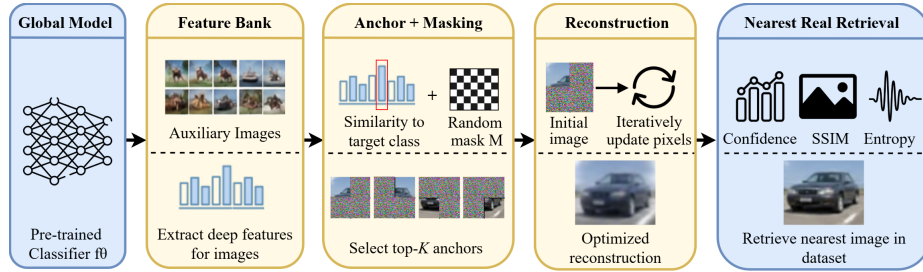


Fig. 2. Overview of the proposed anchor-guided reconstruction framework. Yellow blocks denote the stages of the reconstruction pipeline, while blue blocks denote the trained global model and the final evaluation/retrieval stage.

weak prior rather than as a source of private information. The auxiliary dataset is disjoint from all client training data and serves only to provide semantic initialization and feature-space guidance. Consequently, any successful reconstruction reflects information retained by the trained global model rather than direct access to private client samples. The auxiliary dataset is used to construct the feature bank described in Section 4.

Attack Objective. The adversary’s objective is to reconstruct class-representative inputs by minimizing the multi-objective loss defined in Section 4.

4 Methodology

This section presents the proposed anchor-guided reconstruction framework for FL under the post-hoc threat model defined in Section 3. Given a trained global model f_θ and an auxiliary dataset drawn from the same semantic domain, our goal is to reconstruct class-representative inputs by optimizing directly in the input space.

Our framework consists of three stages: (i) construction of an auxiliary feature bank, (ii) anchor-based initialization via stochastic masking, and (iii) multi-objective optimization-based reconstruction. An overview is illustrated in Figure 2. The attacker constructs a feature bank from auxiliary data and uses it for anchor selection and initialization. The reconstructed inputs are then refined via gradient-based optimization under multiple model-consistency constraints.

4.1 Anchor-Guided Reconstruction Framework

Auxiliary Feature Bank Construction. We construct a feature bank using an auxiliary dataset drawn from the same semantic domain as the FL task. The dataset is independent of all client training data and serves solely as prior knowledge.

Algorithm 1 Anchor-Guided Federated Model Reconstruction

Require: Global model f_θ , feature bank $\mathcal{B}$, target classes $\mathcal{C}$, samples per class N , number of anchors per sample A , optimization steps T , restarts R

Ensure: Reconstructions $\hat{x}$

```

1: for each class  $c \in \mathcal{C}$  do
2:   for each sample index  $i = 1, \dots, N$  do
3:     construct anchor set  $\mathcal{A}_{c,i}$  from  $\mathcal{B}$ 
4:     for each anchor  $x_a \in \mathcal{A}_{c,i}$  do
5:       for each restart  $r = 1, \dots, R$  do
6:         sample binary mask  $M$ 
7:         sample noise  $x_n$ 
8:          $x_0 \leftarrow M \odot x_a + (1 - M) \odot x_n$ 
9:         initialize  $u \leftarrow x_0$ 
10:        for  $t = 1, \dots, T$  do
11:           $x \leftarrow \sigma(u)$ 
12:          compute  $\mathcal{L}(x; f_\theta)$ 
13:           $u \leftarrow u - \eta \nabla_u \mathcal{L}(x)$ 
14:        end for
15:        compute score  $s \leftarrow f_\theta(x)_c$ 
16:        store  $\hat{x}_{c,i,a,r} \leftarrow x$ 
17:      end for
18:    end for
19:    select final reconstruction:
20:     $\hat{x}_{c,i} \leftarrow \arg \max_{x \in \{\hat{x}_{c,i,a,r}\}} f_\theta(x)_c$ 
21:  end for
22: end for

```

Each auxiliary sample x_i is processed using the frozen global model f_θ to extract intermediate and output representations:

$$z_i = \phi_\theta(x_i), \quad p_i = f_\theta(x_i),$$

where $\phi_\theta(\cdot)$ denotes the penultimate-layer feature extractor. The resulting feature bank is defined as:

$$\mathcal{B} = \{(x_i, z_i, p_i)\}_{i=1}^N.$$

This bank encodes semantic class structure and is used for anchor selection during reconstruction.

Partial-Anchor Initialization. To improve optimization stability, reconstruction is initialized using an anchor sample x_a selected from the feature bank. Instead of directly using the anchor, we apply stochastic spatial masking:

$$x_0 = M \odot x_a + (1 - M) \odot x_n,$$

where x_n is randomly sampled noise used only in the masked regions and M is a binary mask selecting the regions preserved from anchor x_a . This preserves partial semantic content while introducing diversity for optimization. We optimize a reparameterized variable u , initialized as $u_0 = x_0$, such that $x = \sigma(u)$ maps the latent variable to the image space.

Multi-Objective Reconstruction. Given initialization x_0 , reconstruction is formulated as:

$$\hat{x} = \arg \min_x \mathcal{L}(x; f_\theta),$$

where the objective combines multiple model-consistency and prior constraints:

$$\mathcal{L}(x) = \mathcal{L}_{ce} + \lambda_{bn}\mathcal{L}_{bn} + \lambda_f\mathcal{L}_{feat} + \lambda_p\mathcal{L}_{perc} + \lambda_{tv}\mathcal{L}_{tv} + \lambda_c\mathcal{L}_{color} + \lambda_a\mathcal{L}_{anchor}.$$

Here, $\mathcal{L}_{ce}$, $\mathcal{L}_{bn}$, $\mathcal{L}_{feat}$, $\mathcal{L}_{perc}$, $\mathcal{L}_{tv}$, $\mathcal{L}_{color}$, and $\mathcal{L}_{anchor}$ denote classification, Batch Normalization, feature consistency, perceptual, total variation, color regularization, and anchor preservation losses, respectively; λ_{bn} , λ_f , λ_p , λ_{tv} , λ_c , and λ_a control their relative contributions.

4.2 Optimization Procedure

Optimization is performed via gradient descent in the input (reparameterized) space. At iteration t , the input is updated as:

$$x^t = \sigma(u^t), \quad u^{t+1} = u^t - \eta \nabla_{u^t} \mathcal{L}(x^t).$$

where $\sigma(\cdot)$ maps the optimization variable to valid image space. Multiple random restarts are employed to mitigate local minima sensitivity. At each restart, optimization is performed independently, and the final reconstruction is selected based on maximum target-class confidence under f_θ . The full procedure, including anchor selection, initialization, optimization, and restart-based selection, is summarized in Algorithm 1. The reconstruction requires $\mathcal{O}(|C|NART)$ optimization iterations, where $|C|$, N , A , R , and T denote the numbers of classes, samples, anchors, restarts, and optimization steps, respectively; thus, the cost scales linearly with each parameter.

5 Experimental Results

We evaluate four heterogeneity regimes: E1 (IID), E2 (mild non-IID), E3 (strong non-IID), and E4 (extreme non-IID). For each setting, we reconstruct samples from all CIFAR-10 classes using multiple anchor positions, yielding a total of 100 reconstructions per experiment. We evaluate reconstructions using MSE (lower is better), SSIM (higher is better), feature distance (lower indicates greater feature similarity), and Unique Nearest (lower indicates stronger concentration around fewer training samples). All settings use the same CIFAR-adapted ResNet-34 and training configuration: 10 clients, 100 rounds, 3 local epochs, batch size 128, SGD ($\eta = 0.05$, momentum = 0.9, weight decay = 5×10^{-4}), and 60% client participation. Only data partitioning varies across E1–E4 ($\alpha = \infty, 0.5, 0.3, 0.1$), with no setting-specific tuning. The federated models used for reconstruction achieved classification accuracies of 94.11%, 93.18%, 89.51%, and 65.39% under E1–E4, respectively. To support reproducibility, the complete implementation, experimental configurations, and analysis scripts are publicly available at: https://github.com/imhamzasajjad/anchor_guided_fl_reconstruction.

Table 1. Results are reported as mean $\pm$ standard deviation (SD), together with the corresponding 95% confidence intervals (CI) of the estimated mean ($N = 100$).

Setting	Metric	Mean $\pm$ SD	95% CI
E1	MSE	0.01323 ± 0.00556	[0.01213, 0.01433]
	SSIM	0.69808 ± 0.09616	[0.67900, 0.71716]
	Feature Distance	0.31286 ± 0.08546	[0.29590, 0.32982]
	Unique Nearest	1.55 ± 0.95	[1.36, 1.74]
E2	MSE	0.01434 ± 0.00750	[0.01285, 0.01583]
	SSIM	0.70477 ± 0.10810	[0.68332, 0.72622]
	Feature Distance	0.36841 ± 0.14018	[0.34060, 0.39623]
	Unique Nearest	1.59 ± 1.00	[1.39, 1.79]
E3	MSE	0.01011 ± 0.00577	[0.00896, 0.01125]
	SSIM	0.74033 ± 0.10346	[0.71980, 0.76086]
	Feature Distance	0.22329 ± 0.07368	[0.20867, 0.23791]
	Unique Nearest	1.20 ± 0.49	[1.10, 1.30]
E4	MSE	0.03111 ± 0.01523	[0.02809, 0.03413]
	SSIM	0.53105 ± 0.14430	[0.50242, 0.55969]
	Feature Distance	0.17414 ± 0.08665	[0.15694, 0.19133]
	Unique Nearest	1.76 ± 0.83	[1.60, 1.92]

5.1 Reconstruction Fidelity and Leakage Trends

Across all settings, reconstruction fidelity varies significantly with data heterogeneity. Table 1 reports reconstruction consistency metrics across heterogeneity settings, including the mean, standard deviation, and the corresponding 95% confidence intervals (CI) of the estimated mean performance.

Strong non-IID setting (E3) achieves the lowest Mean Squared Error (MSE) (0.01011) and the highest Structural Similarity Index (SSIM) (0.74033), indicating superior pixel-level accuracy and structural preservation compared to other settings. The reduced feature distance further suggests that reconstructions in E3 are semantically closer to their original inputs. In contrast, extreme non-IID (E4) leads to degraded pixel-level quality, suggesting unstable optimization dynamics. Leakage-level classification further supports this trend, with E3 exhibiting the highest proportion of high-leakage reconstructions, while E4 shifts toward medium and low leakage cases.

To facilitate interpretation, we categorize reconstruction groups into leakage levels using anchor-consistency statistics derived from multiple anchor-guided reconstructions of the same target class. A reconstruction group is classified as *High Leakage* when the mean pairwise SSIM among generated samples is at least 0.70, the number of unique nearest-neighbour training images is at most two, and the mean pairwise feature distance does not exceed 1.00. A reconstruction group is classified as *Medium Leakage* when the mean pairwise SSIM is at least 0.45,

the number of unique nearest-neighbour training images is at most three, and the mean pairwise feature distance does not exceed 1.60. All remaining groups are classified as *Low Leakage*. These thresholds are applied uniformly across all heterogeneity settings. The threshold values were selected empirically through exploratory analysis of the anchor-consistency distributions and visual inspection of reconstructed samples. Their purpose is not to define universal privacy categories, but rather to provide a consistent and interpretable framework for comparing leakage behaviour across federated heterogeneity regimes. Leakage-level classification in Table 2 further supports these observations. E3 exhibits the highest proportion of high-leakage reconstructions (66%), indicating stronger semantic concentration and more stable reconstructions under strong non-IID conditions. In contrast, E4 shifts toward medium and low leakage (54% and 36%, respectively), suggesting that extreme heterogeneity disrupts optimization stability and reduces the highly confident reconstructions.

Table 2. Leakage-level distribution (%) across heterogeneity settings. Each entry represents the proportion of reconstructed samples classified as high, medium, or low leakage based on reconstruction consistency metrics.

Setting	High (%)	Medium (%)	Low (%)
E1	52.0	42.0	6.0
E2	48.0	42.0	10.0
E3	66.0	34.0	0.0
E4	10.0	54.0	36.0

5.2 Statistical Significance of Heterogeneity Effects

To assess whether the observed differences across heterogeneity regimes are statistically meaningful, we perform Welch’s t-tests on per-sample reconstruction metrics, specifically SSIM and feature distance. These metrics capture both pixel-level similarity and semantic consistency of reconstructions. As shown in Table 3, E3 exhibits statistically significant differences compared to both E1 and E2 in SSIM and feature distance. We use a significance threshold of $p < 0.05$. In particular, higher SSIM and lower feature distance in E3 indicate that reconstruction quality improvements under strong non-IID conditions are unlikely to be due to random sampling variation. These results provide statistical support for the claim that data heterogeneity significantly influences reconstruction stability and fidelity.

5.3 Class-wise Sensitivity Analysis

We further analyze reconstruction sensitivity at the class level to understand how semantic content influences privacy leakage. Results indicate that leakage is

Table 3. Welch’s t-test comparing E3 against E1 and E2 using per-sample SSIM and feature distance ($N = 100$ per setting).

Comparison	Metric	p-value
E3 vs E1	SSIM	0.004
E3 vs E1	Feature Distance	< 0.001
E3 vs E2	SSIM	0.018
E3 vs E2	Feature Distance	< 0.001

strongly class-dependent and varies across heterogeneity settings. In E3, classes such as aeroplane and deer exhibit higher reconstruction variability, while horse remains stable. Under extreme non-IID conditions (E4), reconstruction performance varies across classes, with bird exhibiting high feature distance. These trends suggest that reconstruction vulnerability is jointly determined by class semantics and data partitioning, rather than heterogeneity alone.

Table 4. Hardest and easiest classes under each setting, measured by mean feature distance ($N = 100$ per setting).

Setting	Hardest	Feature Distance	Easiest	Feature Distance
E1	bird	0.4145	ship	0.2561
E2	dog	0.6890	cat	0.2609
E3	airplane	0.3012	horse	0.1664
E4	bird	0.3046	frog	0.0873

5.4 Ablation Study

In order to assess the contribution of the important parts of the proposed framework, we consider anchor-based initialization and the use of feature banks in our experiments. The results of experiments performed for each heterogeneity case are provided in Table 5. Random initialization represents the conventional optimization strategy adopted by existing optimization-based inversion methods, where reconstruction begins from noise without semantic priors. Compared with this baseline, the proposed anchor-guided framework substantially improves reconstruction quality across all heterogeneity settings. Furthermore, removing the feature bank consistently degrades performance, demonstrating that both anchor-guided initialization and auxiliary feature guidance contribute to the overall effectiveness of the proposed framework.

5.5 Qualitative Reconstruction Results

Figure 3 presents qualitative reconstructions across heterogeneity settings. Reconstructions consistently preserve class semantics and remain stable across different anchor positions, indicating strong convergence behavior of the global

Table 5. Ablation study across heterogeneity settings comparing random initialization, anchor-based reconstruction without feature-bank guidance, and the full anchor-guided framework.

Experiment	Method	MSE	SSIM	Feature Distance	Unique Nearest
E1	Random Init	0.0628	0.0826	1.5128	1.00
E1	No Feature Bank	0.0655	0.1194	2.8718	2.56
E1	Full Method	0.0309	0.5166	1.5942	1.42
E2	Random Init	0.0678	0.1041	1.6942	1.00
E2	No Feature Bank	0.0663	0.1205	3.0111	2.52
E2	Full Method	0.0349	0.4963	1.6505	1.48
E3	Random Init	0.0514	0.0907	1.4547	1.00
E3	No Feature Bank	0.0549	0.1794	2.5508	2.08
E3	Full Method	0.0226	0.6385	1.4755	1.22
E4	Random Init	0.0609	0.0834	1.9147	1.00
E4	No Feature Bank	0.0626	0.1103	2.2731	1.66
E4	Full Method	0.0582	0.1369	1.9941	1.70

model. These qualitative results are consistent with quantitative observations, indicating that federated models retain recoverable semantic information even under heterogeneous training conditions.

6 Discussion

This section interprets the empirical findings and examines how data heterogeneity influences privacy leakage in FL under a post-hoc reconstruction threat model. Rather than treating reconstruction quality as an absolute metric, we analyze it as an indicator of the structure and stability of representations learned by the global model. Throughout this paper, privacy leakage is operationalized through reconstruction fidelity and semantic consistency metrics. These measures do not directly quantify information-theoretic privacy loss; rather, they serve as empirical indicators of recoverable information retained within the trained global model. Consequently, stronger reconstruction performance is interpreted as evidence of increased leakage under the considered threat model.

Non-monotonic effect of data heterogeneity. A key observation from our experiments is that privacy leakage exhibits a non-monotonic relationship with data heterogeneity. Across our experiments, the strongest reconstruction performance is observed under strong non-IID conditions (E3), rather than under IID (E1), mildly non-IID (E2), or extreme non-IID (E4) regimes. In this setting, the global model appears to learn representations that are both stable and highly class-consistent, which in turn makes them more amenable to inversion through gradient-based reconstruction. In contrast, IID and mildly heterogeneous settings produce comparatively weaker reconstruction signals, suggesting less structured feature alignment across classes. This indicates that neither

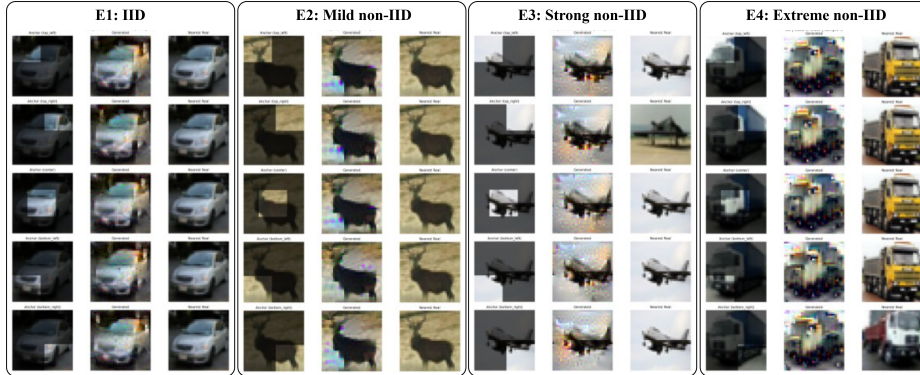


Fig. 3. Representative anchor-guided reconstructions across four data-heterogeneity settings (E1–E4). Each panel shows five anchor locations and their corresponding reconstructions and nearest CIFAR-10 training samples. As different classes are shown across settings, the figure illustrates robustness to anchor placement rather than a controlled class-wise comparison.

fully homogeneous nor highly fragmented data distributions maximize vulnerability; instead, an intermediate regime induces the most favorable conditions for reconstruction-based attacks.

Reconstruction stability and leakage. Our results associate stronger leakage with more consistent reconstructions, as reflected by higher SSIM, lower feature distance, and the larger proportion of high-leakage cases observed in E3. In E3, higher SSIM and lower feature distance indicate that the optimization converges to more structurally and semantically consistent reconstructions. The proposed reconstruction framework amplifies this effect by leveraging auxiliary feature priors and anchor-based initialization. This approach limits optimization to well-structured areas of the learned manifold, thus guaranteeing stable and semantically consistent reconstructions.

Effect of extreme heterogeneity. When facing extremely non-IID conditions (E4), there is a significant loss of reconstruction performance in both pixel-level similarity and structural consistency. However, this degradation does not indicate improved privacy. The federated model trained under E4 also exhibits a substantial reduction in classification accuracy relative to the other heterogeneity settings, suggesting weaker representational capacity and reduced model utility. Therefore, we can assume that this performance decline is most likely associated with poor feature learning rather than information irrecoverability. It should be noted that feature-space similarity remains relatively strong, indicating that semantic information is still partially preserved even when visual quality deteriorates.

Class-dependent vulnerability. We further observe that vulnerability due to reconstruction is highly diverse across the different types of semantic classes. Feature-space reconstruction quality varies across classes and heterogeneity settings, indicating class-dependent sensitivity to data partitioning. These differences suggest that class semantics interact with data partitioning strategies to shape privacy leakage patterns. In extreme non-IID regimes, this imbalance becomes more pronounced, with certain classes degrading more severely than others. This indicates that heterogeneity amplifies existing disparities in representation quality across categories.

Implications for federated learning security. Overall, our findings show that increasing data heterogeneity does not monotonically reduce reconstruction leakage under the evaluated post-hoc attack. Within our experimental setting, strong non-IID training (E3) yields the highest overall reconstruction fidelity, although individual metrics do not always rank all heterogeneity levels identically. This has important implications for privacy evaluation in FL. In particular, benchmarking should not rely solely on IID or highly skewed settings, as these may underestimate the true privacy risk. Even under strict post-hoc constraints where the adversary has access only to the final global model, significant information leakage persists through learned representations.

These results highlight the need for stronger privacy mechanisms that explicitly account for information encoded in model parameters, rather than focusing solely on training-time leakage channels.

7 Limitations and Future Work

While the proposed anchor-guided reconstruction framework demonstrates consistent performance across heterogeneous federated settings, we highlight several controlled design choices that define the scope of this study and suggest directions for future extensions.

First, our evaluation is conducted on CIFAR-10, which provides a standard benchmark for FL and reconstruction-based attacks. Although CIFAR-10 provides a controlled benchmark for systematically studying the effect of data heterogeneity, validating whether the observed non-monotonic leakage pattern generalizes to more complex datasets remains an important direction for future work. Extending the analysis to datasets such as CIFAR-100, Tiny-ImageNet, or domain-specific datasets would help assess the generality of the observed behavior. Second, the proposed reconstruction framework relies on an auxiliary feature bank constructed from publicly available data drawn from the same semantic domain as the target task. While consistent with prior representation-guided inversion methods, this introduces a domain-level prior that may not be available in data-scarce settings. Exploring alternatives that remove or learn this prior in a data-free setting remains an open direction for future work. Finally, the observed heterogeneity-leakage relationship is specific to the considered reconstruction framework and metrics; evaluating its generality across alternative post-hoc attacks, privacy measures, and model families remains future work.

8 Conclusion

We studied post-hoc data reconstruction attacks in FL under varying degrees of data heterogeneity, focusing on a strict threat model where the adversary has access only to the final global model. We proposed an anchor-guided reconstruction framework that combines partial-anchor initialization with auxiliary feature priors to improve inversion quality without requiring access to client updates or training trajectories. Ablation results confirm that anchor initialization and feature-bank guidance both contribute substantially to reconstruction fidelity and semantic consistency. Under the evaluated attack and metrics, reconstruction exhibits a non-monotonic trend: strong non-IID training yields the highest reconstruction fidelity, whereas extreme non-IID training reduces reconstruction effectiveness alongside substantially lower model accuracy. This research therefore shows that federated models can still retain much of the information about their training data even when working in realistic non-IID settings.

9 Acknowledgments

This work was partially supported by the RDS PTR 25–27 Project 2.1 Cybersecurity and by SPF BOSA through the “Financement de projets pour le développement de l’intelligence artificielle en Belgique” AI Call 2023 project *AIDE* (Federated Learning for Robust, Resilient and Adaptive Protection of Systems).

References

1. Bai, F., Zhao, Y., Shen, T., Zeng, K., Zhang, X., Zhang, C.: Fedopcs: An optimized poisoning countermeasure for non-iid federated learning with privacy-preserving stability. *Symmetry* **17**(5), 782 (2025). <https://doi.org/10.3390/SYM17050782>, <https://doi.org/10.3390/sym17050782>
2. Bai, L., Hu, H., Ye, Q., Li, H., Wang, L., Xu, J.: Membership inference attacks and defenses in federated learning: A survey. *ACM Comput. Surv.* **57**(4) (Dec 2024)
3. Bertran, M.A., Tang, S., Kearns, M., Morgenstern, J.H., Roth, A., Wu, S.: Reconstruction attacks on machine unlearning: Simple models are vulnerable. In: The Thirty-eighth Annual Conference on Neural Information Processing Systems (2024), <https://openreview.net/forum?id=i4gqCM1r3z>
4. Bharati, S., Mondal, M.R.H., Podder, P., Prasath, V.S.: Federated learning: Applications, challenges and future directions. *Int. J. Hybrid Intell. Syst.* **18**(1–2), 19–35 (Jan 2022). <https://doi.org/10.3233/HIS-220006>, <https://doi.org/10.3233/HIS-220006>
5. Dimitrov, D.I., Baader, M., Müller, M.N., Vechev, M.: Spear: Exact gradient inversion of batches in federated learning. In: *Advances in Neural Information Processing Systems (NeurIPS)*. vol. 37 (2024)
6. Fang, L., Wang, L., Li, H.: Iterative and mixed-spaces image gradient inversion attack in federated learning. *Cybersecurity* **7**(35) (2024)

7. Ge, L., Li, H., Wang, X., Wang, Z.: A review of secure federated learning: Privacy leakage threats, protection technologies, challenges and future directions. *Neuro-computing* **561**, 126897 (2023). <https://doi.org/https://doi.org/10.1016/j.neucom.2023.126897>, <https://www.sciencedirect.com/science/article/pii/S0925231223010202>
8. Geiping, J., Bauermeister, H., Dröge, H., Moeller, M.: Inverting gradients – how easy is it to break privacy in federated learning? In: *Advances in Neural Information Processing Systems*. vol. 33, pp. 16937–16947 (2020)
9. Haldankar, A., Riasi, A., Nguyen, H.D., Phuong, T., Hoang, T.: Breaking privacy in model-heterogeneous federated learning. In: *Proceedings of the 27th International Symposium on Research in Attacks, Intrusions and Defenses*. p. 465–479. RAID '24, Association for Computing Machinery, New York, NY, USA (2024). <https://doi.org/10.1145/3678890.3678905>, <https://doi.org/10.1145/3678890.3678905>
10. Hong, H.G., Cho, Y., Cho, H., et al.: Foreseeing reconstruction quality of gradient inversion: An optimization perspective. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. vol. 38, pp. 12473–12481 (2024)
11. Hsu, T.M.H., Qi, Brown, M.: Measuring the effects of non-identical data distribution for federated visual classification. *ArXiv abs/1909.06335* (2019), <https://api.semanticscholar.org/CorpusID:202572978>
12. Karimireddy, S.P., Kale, S., Mohri, M., Reddi, S., Stich, S., Suresh, A.T.: SCAF-FOLD: Stochastic controlled averaging for federated learning. In: III, H.D., Singh, A. (eds.) *Proceedings of the 37th International Conference on Machine Learning*. vol. 119, pp. 5132–5143. PMLR (2020)
13. Konečný, J., McMahan, H.B., Ramage, D., Richtárik, P.: Federated optimization: Distributed machine learning for on-device intelligence. *ArXiv* (2016), <https://api.semanticscholar.org/CorpusID:2549272>
14. Li, Q., Wen, Z., Wu, Z., Hu, S., Wang, N., Li, Y., Liu, X., He, B.: A survey on federated learning systems: Vision, hype and reality for data privacy and protection. *IEEE Transactions on Knowledge and Data Engineering* **35**(4), 3347–3366 (2023). <https://doi.org/10.1109/TKDE.2021.3124599>
15. Li, T., Sahu, A.K., Zaheer, M., Sanjabi, M., Talwalkar, A., Smith, V.: Federated optimization in heterogeneous networks (2020), <https://arxiv.org/abs/1812.06127>
16. Liu, J., Chen, B., Xue, B., Guo, M., Xu, Y.: Piafgnn: Property inference attacks against federated graph neural networks. *Computers, Materials and Continua* **82**(2), 1857–1877 (2025)
17. McMahan, H.B., Moore, E., Ramage, D., Hampson, S., y Arcas, B.A.: Communication-efficient learning of deep networks from decentralized data (2023), <https://arxiv.org/abs/1602.05629>
18. Melis, L., Song, C., De Cristofaro, E., Shmatikov, V.: Exploiting unintended feature leakage in collaborative learning. In: *IEEE Symposium on Security and Privacy*. pp. 691–706 (2019). <https://doi.org/10.1109/SP.2019.00029>
19. Pustozero, A., Mayer, R.: Information leaks in sequential federated learning. In: *Proceedings of the Workshop on Decentralized IoT Systems and Security (DISS) 2020*. The Internet Society, San Diego, CA, USA (2020). <https://doi.org/10.14722/diss.2020.23004>, <https://dx.doi.org/10.14722/diss.2020.23004>
20. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. *IEEE Symposium on Security and Privacy* pp. 3–18 (2016)

21. Sun, T., Li, D., Wang, B.: Decentralized federated averaging. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **45**(4), 4289–4301 (2023). <https://doi.org/10.1109/TPAMI.2022.3196503>
22. Tariq, A., Serhani, M.A., Sallabi, F.M., Barka, E.S., Qayyum, T., Khater, H.M., Shuaib, K.A.: Trustworthy federated learning: A comprehensive review, architecture, key challenges, and future research prospects. *IEEE Open Journal of the Communications Society* **5**, 4920–4998 (2024). <https://doi.org/10.1109/OJCOMS.2024.3438264>
23. Wang, F., Hugh, E., Li, B.: More than enough is too much: Adaptive defenses against gradient leakage in production federated learning. In: *IEEE INFOCOM* (2023)
24. Yin, H., Mallya, A., Vahdat, A., Alvarez, J.M., Kautz, J., Molchanov, P.: See through gradients: Image batch recovery via gradinversion. In: *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 16332–16341 (2021). <https://doi.org/10.1109/CVPR46437.2021.01607>
25. Yin, H., Molchanov, P., Li, Z., Alvarez, J.M., Mallya, A., Hoiem, D., Jha, N.K.: Dreaming to distill: Data-free knowledge transfer via deepinversion. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. pp. 8715–8724 (2020). <https://doi.org/10.1109/CVPR42600.2020.00874>
26. Yurdem, B., Kuzlu, M., Gullu, M.K., Catak, F.O., Tabassum, M.: Federated learning: Overview, strategies, applications, tools and future directions. *Heliyon* **10**(19), e38137 (2024). <https://doi.org/https://doi.org/10.1016/j.heliyon.2024.e38137>, <https://www.sciencedirect.com/science/article/pii/S2405844024141680>
27. Zhao, Y., Li, M., Lai, L., Suda, N., Civin, D., Chandra, V.: Federated learning with non-iid data (2018)
28. Zhu, H., Xu, J., Liu, S., Jin, Y.: Federated learning on non-iid data: A survey. *Neurocomputing* **465**, 371–390 (2021). <https://doi.org/https://doi.org/10.1016/j.neucom.2021.07.098>, <https://www.sciencedirect.com/science/article/pii/S0925231221013254>
29. Zhu, L., Liu, Z., Han, S.: Deep leakage from gradients. In: Wallach, H., Larochelle, H., Beygelzimer, A., d’Alché Buc, F., Fox, E., Garnett, R. (eds.) *Advances in Neural Information Processing Systems* 32. pp. 14774–14784 (2019)

Privacy-Preserving Decision Tree Inference under CKKS: A Precision Impact Analysis

Madicke-Diadji Mbodj^{1,2}, Mawloud Omar², Reda Yaich¹, Anis Bkakria¹, and
Siham Bouchelaghem²

¹ IRT SystemX, Palaiseau, France

madicke-diadji.mbodj@irt-systemx.fr, reda.yaich@irt-systemx.fr,
anis.bkakria@irt-systemx.fr

² Univ. Bretagne Sud, IRISA, UMR 6074, Vannes, France

mawloud.omar@univ-ubs.fr, siham.bouchelaghem@univ-ubs.fr

Abstract. Privacy-preserving machine learning inference via homomorphic encryption (HE) enables a server to evaluate a model on encrypted client data without accessing plaintext inputs. Decision trees are attractive for this setting, but their discontinuous comparisons and data-dependent branches are difficult to evaluate with the arithmetic supported by CKKS. We study this problem by combining the existing HBDT-SUMPATH traversal with SOFTSTEP polynomial comparisons.

We introduce a Soft-Adaptive degree-assignment strategy in which each internal node receives a polynomial degree according to its calibrated decision difficulty. We further provide a precision-impact analysis that separates polynomial-approximation loss from additional CKKS-induced loss and bounds their propagation through SUMPATH scores. The end-to-end implementation uses Microsoft SEAL v4.1.2 at 128-bit security. The evaluation covers seven single-tree benchmarks and a random-forest extension on the cybersecurity workloads Road and Website Phishing. Adaptive HE matches adaptive cleartext on all retained single-tree runs; the remaining losses on Iris, Wine, and Website Phishing are therefore attributable to polynomial approximation. Road RF and Website Phishing RF preserve their retained cleartext accuracy under encrypted evaluation. These results show that precision depends strongly on workload and decision margins, while the forest experiments provide an initial scalability stress test.

Keywords: Homomorphic Encryption · CKKS · Decision Tree · Random Forest · Privacy-Preserving Inference · Polynomial Approximation · Data Privacy

1 Introduction

The growing adoption of Machine Learning as a Service (MLaaS) raises a fundamental privacy challenge: clients often submit sensitive inputs to a service provider in order to obtain predictions. In healthcare, finance, mobility, and cybersecurity, these inputs may contain confidential or regulated information. Pri-

vate inference addresses this problem by enabling a server to evaluate a model over encrypted inputs without learning the client’s query.

Decision trees are compact, interpretable, and widely used, but their threshold comparisons and data-dependent branches are difficult to evaluate under homomorphic encryption (HE). This limitation is particularly visible under CKKS, which efficiently supports approximate additions, multiplications, rescaling, and rotations over encrypted vectors, but not native comparison or branching. A test such as $x_f \leq \theta$ must therefore be replaced by an arithmetic surrogate, introducing polynomial-approximation error in addition to the numerical error of CKKS arithmetic.

This paper studies the interaction between these two error sources. We build on the HBDT-SUMPATH traversal of Shin et al. [20], which reduces tree traversal to arithmetic aggregation of node indicators, and on the SOFTSTEP comparator of Akavia et al. [1]. Our key observation is that internal nodes have different decision difficulties: nodes whose samples remain far from the threshold can use low-degree approximations, while nodes close to a decision boundary require higher accuracy.

This paper provides the following contributions.

- **Soft-Adaptive degree assignment.** We introduce a node-wise degree-selection rule that assigns the polynomial degree according to the calibrated minimum feature gap, using low degrees for stable nodes and higher degrees for sensitive nodes.
- **Precision-impact analysis.** We separate accuracy loss caused by polynomial approximation from additional loss induced by encrypted CKKS evaluation and analyse how node-level errors propagate through the SUMPATH mismatch score.
- **End-to-end proof-of-concept evaluation.** We compare hard cleartext, Soft-Adaptive cleartext, and encrypted Soft-Adaptive inference on seven single-tree benchmarks and extend the evaluation to small random forests on two cybersecurity workloads.
- **Comparative evaluation.** We compare the proposed approach with representative privacy-preserving decision-tree frameworks in terms of comparison mechanism, traversal strategy, encryption setting, accuracy, runtime, and precision–efficiency trade-offs.

The remainder of the paper is organized as follows. Section 2 introduces decision trees, CKKS, SOFTSTEP, and SUMPATH. Section 3 reviews related work. Section 4 presents the Soft-Adaptive method and discusses its scalability to forests. Section 5 develops the precision-impact analysis. Section 6 reports the experiments, and Section 7 concludes the paper.

2 Background

This section introduces the existing technical building blocks used by our approach. In particular, SOFTSTEP polynomial comparison and HBDT-SUMPATH

traversal are prior mechanisms proposed respectively by Akavia et al. and Shin et al. They are not contributions of this paper; rather, they provide the comparison and traversal foundations on which our Soft-Adaptive degree assignment and precision-impact analysis are built.

2.1 Binary Decision Trees

A binary decision tree T classifies an input $x \in \mathbb{R}^{d_{\text{feat}}}$ by routing it from the root to a leaf. Each internal node N_i stores a feature index f_i and threshold θ_i and evaluates

$$c_i(x) = \mathbf{1}[x_{f_i} \leq \theta_i]. \quad (1)$$

The left branch is selected when $c_i(x) = 1$, and the right branch otherwise. Under HE, the server cannot branch on $c_i(x)$ while it remains encrypted. Private tree evaluation must therefore replace conditional control flow by an exact encrypted protocol or by an arithmetic formulation.

2.2 CKKS and SIMD Arithmetic

CKKS [5] is a public key homomorphic encryption scheme that encodes approximate real or complex values into ciphertexts over a polynomial ring. It supports additions, multiplications, rescaling, and slot rotations. Its *Single Instruction, Multiple Data* (SIMD) capability means that one ciphertext contains a vector of up to $N/2$ complex slots, where N is the polynomial-modulus degree. Homomorphic additions and multiplications act component-wise on all slots, while rotations permute the packed values and enable aggregation across slots.

The main precision-performance parameters are the polynomial modulus degree N , the scale $\Delta = 2^s$, and the coefficient-modulus chain. Multiplication followed by rescaling consumes one modulus level and introduces a small rounding error. A simplified level-wise model is

$$\varepsilon_\ell \approx \ell \sqrt{N} \Delta^{-1}, \quad (2)$$

where ℓ is the number of consumed multiplicative levels. The practical question is not whether CKKS is exact, but whether its residual error is small compared with the decision and leaf-score margins of the application.

Our implementation uses vertical packing. Features from one or several samples occupy contiguous slot blocks; the same encrypted arithmetic is then applied to several packed samples. This amortises polynomial evaluation and key-switching costs, although rotations and unused slots can limit efficiency for small batches.

2.3 SoftStep Polynomial Comparison

The Heaviside function is discontinuous and cannot be uniformly approximated with low error by a low-degree polynomial on an interval containing the discontinuity. SoftStep [1] excludes a neglected window $(-\delta, \delta)$ around the threshold.

Table 1. Representative SoftStep error and BSGS depth for $\delta = 0.05$.

Degree d	4	8	16	32
$\varepsilon_{\text{poly}}(d, \delta)$	≈ 0.15	≈ 0.06	≈ 0.02	< 0.005
BSGS multiplicative depth ℓ_d	2	3	4	5

Let $I_0(t) = \mathbf{1}[t \geq 0]$. A degree- d polynomial ϕ_d is fitted on a normalised interval, e.g., $[-2, 2]$, by minimising

$$\phi_d = \arg \min_{p \in \mathcal{P}_d} \int_{-2}^2 (I_0(t) - p(t))^2 w_\delta(t) dt, \quad w_\delta(t) = \mathbf{1}[|t| \geq \delta]. \quad (3)$$

The approximation error outside the neglected window is

$$\varepsilon_{\text{poly}}(d, \delta) = \max_{|t| \geq \delta} |\phi_d(t) - I_0(t)|. \quad (4)$$

Increasing d generally reduces approximation error but increases the number of ciphertext multiplications and the required modulus levels. We evaluate the polynomial using a Baby-Step Giant-Step (BSGS) strategy [11], rather than a naive sequential evaluation.

At an internal node N_i , the hard comparison $c_i(x) = \mathbf{1}[x_{f_i} \leq \theta_i]$ is replaced by evaluating the polynomial on the signed margin $t_i(x) = \theta_i - x_{f_i}$. The corresponding approximate right-branch indicator is

$$I_i(x) = 1 - \phi_{d_i}(t_i(x)). \quad (5)$$

With this convention, $I_i(x) \approx 0$ represents a left decision and $I_i(x) \approx 1$ represents a right decision. The neglected window is therefore the region in which the soft indicator may remain far from its Boolean target and where the final prediction is most sensitive to approximation error.

2.4 HBDT-SumPath Traversal

HBDT-SUMPATH, introduced by Shin et al. [20], removes data-dependent tree traversal by evaluating every candidate leaf arithmetically. Using the approximate right-branch indicator from Eq. (5), consider a leaf L_k with path $P_k = (N_{i_1}, \dots, N_{i_{D_k}})$. Let $b_{k,j} \in \{0, 1\}$ denote the branch expected by that leaf, with 0 for left and 1 for right. The SUMPATH mismatch score is

$$S_{\text{mis}}(L_k) = \sum_{j=1}^{D_k} [b_{k,j}(1 - I_{i_j}) + (1 - b_{k,j})I_{i_j}]. \quad (6)$$

In the Boolean case, the reached leaf has score 0 and every other leaf has score at least 1. The prediction is

$$\hat{k} = \arg \min_k S_{\text{mis}}(L_k). \quad (7)$$

Once all node indicators are available, Eq. (6) is affine: additions, plaintext multiplications by path masks, and rotations are sufficient. Therefore, tree depth increases total work but does not multiply the polynomial-evaluation depth along a root-to-leaf path.

Why HBDT-SumPath? We select HBDT-SUMPATH because it aggregates the encrypted node indicators into a mismatch score for each candidate leaf without requiring data-dependent branching. For a given leaf, each node contribution is either I_i or $1 - I_i$, depending on the expected branch. Consequently, once the SOFTSTEP indicators have been evaluated, the leaf scores are obtained mainly through additions and subtractions, without additional ciphertext-ciphertext multiplications.

This property is particularly important under CKKS, since ciphertext ciphertext multiplications consume multiplicative levels and typically require relinearisation and rescaling, thereby increasing both computational cost and numerical error. As a result, the multiplicative depth of the complete tree evaluation is primarily determined by the most expensive SOFTSTEP polynomial rather than by the number of nodes along a root-to-leaf path. In addition, all candidate leaf scores are evaluated arithmetically, so the server does not learn the sequence of branch decisions. These properties make HBDT-SUMPATH well suited to our Soft-Adaptive CKKS inference pipeline.

2.5 Illustrative SoftStep-SumPath Example

Consider the depth-two tree in Fig. 1. The server evaluates three encrypted SoftStep indicators and obtains, after decryption for illustration only, $I_1 = 0.95$, $I_2 = 0.70$, and $I_3 = 0.10$. SumPath evaluates all leaves:

$$\begin{aligned} S(L_{00}) &= I_1 + I_2 = 1.65, & S(L_{01}) &= I_1 + (1 - I_2) = 1.25, \\ S(L_{10}) &= (1 - I_1) + I_3 = 0.15, & S(L_{11}) &= (1 - I_1) + (1 - I_3) = 0.95. \end{aligned} \quad (8)$$

The selected leaf is L_{10} because it has the minimum mismatch score, which is the closest to zero among all candidate leaves. The server never executes an encrypted `if` statement and never learns whether an intermediate decision is left or right: it computes every encrypted score and returns the score vector to the client.

3 Related Work

Private decision-tree evaluation has been studied using secure multiparty computation, garbled circuits, oblivious transfer, and HE. Exact approaches can provide faithful comparisons but often require interaction or substantial communication [13,22,21]. Under HE, bitwise and TFHE methods implement exact Boolean decisions [10,18]; arithmetic comparison protocols operate on encrypted signed differences [17,2,15]; and combinatorial encodings target amortised comparison cost and batching [6,7].

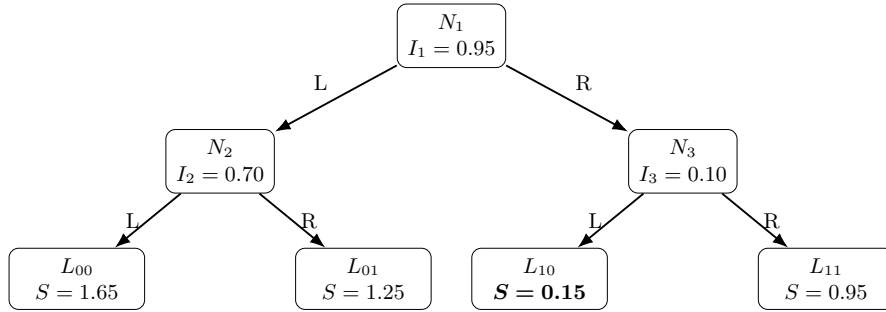


Fig. 1. Illustrative SOFTSTEP-SUMPATH evaluation. All four leaf scores are computed without data-dependent branching; L_{10} is selected because its mismatch score is minimal.

Approximate approaches replace hard comparisons by smooth or polynomial functions. Xiao et al. [23] use a Chebyshev approximation of a steep sigmoid, while Sarpatwar et al. [19] map tree ensembles to neural-network-like structures with soft branches. Akavia et al. [1] introduce the neglected-window SOFTSTEP comparator, and Shin et al. [20] reformulate traversal through HBDT-SUMPATH. These mechanisms are reviewed in Section 2 and are used as building blocks rather than claimed as contributions.

Our work differs in its focus on precision. Instead of using one global polynomial degree, the proposed Soft-Adaptive strategy assigns degrees node-wise from calibrated margins. We then decompose task-level loss into polynomial-approximation and CKKS components and evaluate the same methodology on both single trees and a small random-forest extension. Cross-project results are treated as contextual rather than controlled benchmarks because implementations, models, backends, and execution modes differ.

4 Private Inference with Soft-Adaptive Polynomial Degrees

Building on the SOFTSTEP comparator and HBDT-SUMPATH traversal introduced in Section 2, this section presents the proposed Soft-Adaptive pipeline. The method calibrates each internal node, assigns a polynomial degree according to its local decision difficulty, and evaluates the resulting indicators and leaf scores under CKKS.

4.1 Overview of the Proposed Pipeline

The protocol involves two roles: a client and a server, where the server also acts as the model owner. The server transforms its trained decision tree into a homomorphic-inference compatible form by extracting each node’s feature index

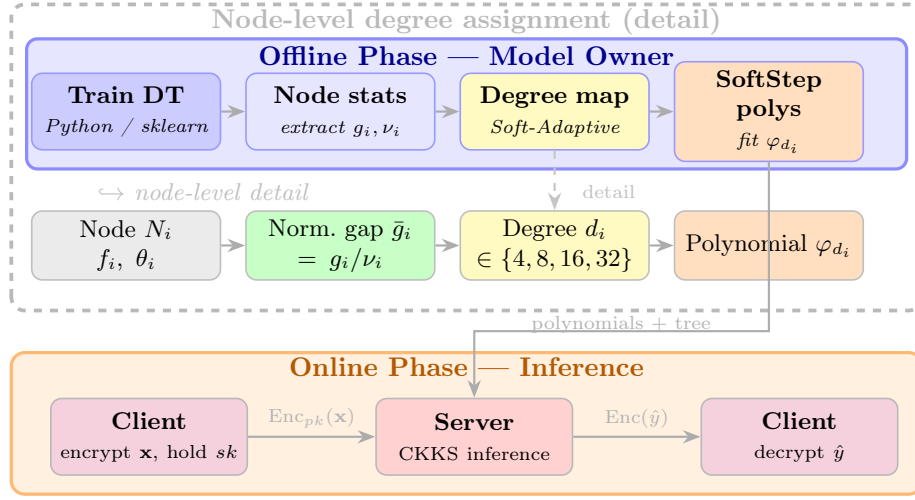


Fig. 2. Soft-Adaptive workflow. Offline, each node receives a polynomial degree from its calibrated feature gap. Online, encrypted SOFTSTEP indicators are computed and aggregated through SUMPATH scores.

and threshold, computing calibration statistics, assigning a polynomial degree, and fitting the corresponding SOFTSTEP polynomial.

At inference time, the client generates the CKKS keys, keeps the secret key private, encrypts the query, and sends the ciphertext together with the public and evaluation keys to the server. The server evaluates the transformed tree on the encrypted query by computing soft node indicators and encrypted SUMPATH scores. The client then decrypts the returned scores and selects the predicted leaf. Figure 2 summarises the resulting Soft-Adaptive workflow, from offline node calibration to online encrypted SUMPATH inference.

4.2 Security Model

We consider a two-party private-inference setting where the server also acts as the model owner. The client owns the private input x , generates the CKKS keys, keeps the secret key private, and sends the encrypted query together with the public and evaluation keys to the server. The server owns the trained tree, transforms it into its homomorphic-inference form, and evaluates it on the encrypted query.

We assume a semi-honest server and rely on the IND-CPA security of CKKS to protect the client's input from the ciphertexts and encrypted intermediate computations. Since the server owns the model, model privacy against the server is not relevant. The privacy of the model against the client, malicious security, hiding the full prediction vector, and verifiable computation are outside the scope of this work.

4.3 Soft-Adaptive Degree Assignment and Calibration

Nodes far from their thresholds can use low-degree approximations, whereas nodes close to a decision boundary require higher accuracy. A single global degree may therefore waste homomorphic depth on stable nodes while remaining insufficient for sensitive ones.

For an internal node N_i , let $X_{\text{calib},i}$ denote the calibration samples reaching that node under hard cleartext traversal. We define the minimum feature gap and the local margin range as

$$g_i = \min_{x \in X_{\text{calib},i}} |x_{f_i} - \theta_i|, \quad \nu_i = \max_{x \in X_{\text{calib},i}} |x_{f_i} - \theta_i|. \quad (9)$$

The quantity g_i measures the smallest distance between the samples reaching node N_i and its threshold: a large value indicates a stable decision, whereas a small value indicates that at least one calibration sample lies close to the decision boundary.

Although the input features are globally scaled to $[-1, 1]$, the signed decision margin

$$t_i(x) = \theta_i - x_{f_i} \quad (10)$$

can still vary substantially across nodes. In particular, even when both x_{f_i} and θ_i belong to $[-1, 1]$, the raw margin may lie in $[-2, 2]$, and its effective range depends on the threshold and on the samples reaching the node. Hence, ν_i does not renormalise the input features; it normalises the local decision margin so that node difficulties can be compared on a common scale.

We therefore define the normalised margin and minimum gap as

$$\tilde{t}_i(x) = \frac{\theta_i - x_{f_i}}{\max(\nu_i, \epsilon_0)}, \quad \bar{g}_i = \frac{g_i}{\max(\nu_i, \epsilon_0)}, \quad (11)$$

where $\epsilon_0 > 0$ avoids numerical instability when ν_i is very small. For the calibration samples reaching N_i , this transformation gives $|\tilde{t}_i(x)| \leq 1$. Thus, $\bar{g}_i$ measures the minimum distance to the threshold relative to the local margin range and provides a scale-independent indicator of node difficulty.

The practical Soft-Adaptive degree map is consequently applied to the normalised gap $\bar{g}_i$:

$$d_i^{\text{raw}} = \begin{cases} 4, & \bar{g}_i \geq 0.08, \\ 8, & 0.04 \leq \bar{g}_i < 0.08, \\ 16, & 0.015 \leq \bar{g}_i < 0.04, \\ 32, & \bar{g}_i < 0.015, \end{cases} \quad d_i^{\text{used}} = \min(d_i^{\text{raw}}, d_{\text{max}}^{\text{profile}}). \quad (12)$$

Stable nodes therefore use low-degree polynomials, whereas nodes whose samples lie relatively close to their thresholds receive higher-degree approximations, subject to the multiplicative-depth budget of the active CKKS profile. All quantities g_i , ν_i , and $\bar{g}_i$ are computed exclusively from training-only or held-out calibration data; test samples are reserved for the final accuracy and runtime evaluation.

4.4 Encrypted Evaluation and Scalability

For each node N_i , the server evaluates the normalised encrypted margin and computes $\tilde{I}_i = 1 - \phi_{d_i^{\text{used}}}(\tilde{t}_i)$. Node polynomials are independent and can be evaluated without following a root-to-leaf control-flow path. The encrypted leaf scores are then obtained from Eq. (6); because the score is affine in the node indicators, this stage requires additions, masks, and rotations but no additional ciphertext–ciphertext multiplication chain proportional to tree depth.

Let M denote the number of internal nodes and $C_{\text{poly}}(d)$ the cost of evaluating a SoftStep polynomial of degree d . Since all internal-node comparisons must be evaluated, the total comparison cost is

$$C_{\text{cmp}} = \sum_{i=1}^M C_{\text{poly}}(d_i^{\text{used}}). \quad (13)$$

However, the node polynomials are independent and can be evaluated in parallel. Hence, the multiplicative depth is primarily determined by the most expensive node polynomial:

$$L_{\text{mult}} \approx \max_{1 \leq i \leq M} L_{\text{poly}}(d_i^{\text{used}}). \quad (14)$$

Once the node indicators are available, SumPath aggregates them into leaf scores using affine operations and therefore introduces no additional ciphertext–ciphertext multiplication depth. The SUMPATH aggregation is bounded by $O(LD)$ node–leaf incidences; larger or deeper trees therefore increase work, rotations, memory, and latency even when multiplicative depth remains governed by the largest polynomial degree.

For a forest of F independently evaluated trees, a first-order cost model is

$$C_{\text{RF}} \approx \sum_{r=1}^F C(T_r), \quad \mathcal{L}_{\text{RF}} \approx \max_r \mathcal{L}(T_r), \quad (15)$$

provided that tree outputs are aggregated without introducing a deeper comparison circuit. The cybersecurity forest experiments in Section 6 therefore serve as an initial ensemble scalability stress test; a controlled study over depth, node count, memory, ciphertext size, and communication is left for future work.

5 Precision Impact Analysis

We next separate the two error sources and study their propagation through SUMPATH.

5.1 Accuracy-Loss Decomposition

Let A_{hard} , A_{soft} , and A_{HE} denote the accuracies of hard cleartext, Soft-Adaptive cleartext, and Soft-Adaptive HE inference. We decompose the total loss as

$$\Delta_{\text{approx}} = A_{\text{hard}} - A_{\text{soft}}, \quad (16)$$

$$\Delta_{\text{CKKS}} = A_{\text{soft}} - A_{\text{HE}}, \quad (17)$$

$$\Delta_{\text{total}} = A_{\text{hard}} - A_{\text{HE}} = \Delta_{\text{approx}} + \Delta_{\text{CKKS}}. \quad (18)$$

The first term measures degradation already caused by replacing hard comparisons with polynomial indicators; the second isolates additional degradation introduced by encrypted approximate arithmetic. This decomposition is central to the experimental analysis below.

5.2 Error Propagation Through SUMPATH

Let $c_i \in \{0, 1\}$ be the exact right-branch indicator. We separate polynomial and encrypted errors as

$$I_i = c_i + e_i^{\text{poly}}, \quad \hat{I}_i^{\text{HE}} = c_i + e_i^{\text{poly}} + e_i^{\text{CKKS}}, \quad (19)$$

with

$$|e_i^{\text{poly}} + e_i^{\text{CKKS}}| \leq \varepsilon_i^{\text{total}} \leq \varepsilon_{\text{poly}}(d_i, \delta) + B_{\text{rs}} \ell_i 2^{-s}, \quad (20)$$

where ℓ_i is the multiplicative depth of the node polynomial and $B_{\text{rs}} \approx N^{1/2}$ models rescaling noise.

Theorem 1 (SUMPATH error bound). *For any leaf L_k with root-to-leaf path $\mathcal{P}_k = (N_{i_1}, \dots, N_{i_D})$, the difference between the encrypted score and the exact Boolean score satisfies*

$$\left| \hat{S}^{\text{HE}}(L_k) - S^{\text{bool}}(L_k) \right| \leq \sum_{j=1}^D \varepsilon_{i_j}^{\text{total}} \leq D \max_i \varepsilon_i^{\text{total}}. \quad (21)$$

Proof. Substituting $\hat{I}_{i_j}^{\text{HE}} = c_{i_j} + e_{i_j}$ into Eq. (6) yields $\hat{S}^{\text{HE}}(L_k) - S^{\text{bool}}(L_k) = \sum_{j=1}^D (1 - 2b_j)e_{i_j}$. Because $b_j \in \{0, 1\}$, $|1 - 2b_j| = 1$; the result follows directly from the triangle inequality and $|e_{i_j}| \leq \varepsilon_{i_j}^{\text{total}}$.

In the Boolean case, the reached leaf has score 0 and any incorrect leaf has score at least 1. A sufficient condition for preserving the selected leaf is therefore

$$D \max_i \varepsilon_i^{\text{total}} < \frac{1}{2}, \quad \varepsilon_{\text{poly}}(d_i, \delta) + B_{\text{rs}} \ell_i 2^{-s} < \frac{1}{2D}. \quad (22)$$

Table 2. Precision regimes as a function of the minimum feature gap.

Regime	Condition	Dominant risk	Mitigation
I: safe	$\text{min_gap} \geq \delta$	negligible loss	low degree
II: approx.	$\varepsilon \ll \text{min_gap} < \delta$	approximation loss	increase degree
III: critical	$\text{min_gap} \ll \delta$	weak score separation	calibrate, raise precision, retrain

5.3 Precision Regimes

The behaviour of the pipeline is governed by the relation between the minimum feature gap min_gap , the neglected-window width δ , and the effective CKKS noise level ε . Table 2 summarises the three main regimes.

Regime I corresponds to samples outside the neglected window, where both approximation and CKKS losses are expected to be negligible. In Regime II, some samples enter the neglected window and cleartext polynomial approximation becomes the dominant risk. Regime III contains samples very close to a threshold; soft indicators can remain near $1/2$, making leaf scores weakly separated and potentially sensitive to both approximation and residual CKKS noise. Mitigation may require stronger node-wise normalisation, increased precision, or retraining with larger decision margins.

5.4 Optimal Degree Assignment

From Eq. (22), the minimum-cost degree at node N_i can be written as

$$d_i^* = \min \left\{ d \in \{4, 8, 16, 32\} : \varepsilon_{\text{poly}}(d, \delta) < \min \left(\frac{1}{2D} - \eta_i, \frac{g_i}{2} \right) \right\}, \quad (23)$$

where

$$\eta_i = B_{\text{rs}} \ell_i 2^{-s}$$

is the node-level CKKS noise term. Since $\varepsilon_{\text{poly}}(d, \delta)$ decreases with d , while the depth cost ℓ_d increases with d , the goal is to choose the smallest degree that provides sufficient approximation accuracy at the node. The practical thresholds in Eq. (12) instantiate this rule in a discrete Soft-Adaptive degree map.

6 Experimental Evaluation

This section evaluates the proposed Soft-Adaptive CKKS inference pipeline. We first describe the experimental setup and benchmark scope, then report the main accuracy and precision-decomposition results, and finally compare our implementation with representative privacy-preserving decision-tree frameworks.

Table 3. Runtime HE profiles used in the implementation.

Profile	Feature range	$n_{\max}$	Max adapt. deg.
Small	$d \leq 4$	4	32
Medium	$5 \leq d \leq 16$	2	16
Large	$d > 16$	2	8

6.1 Experimental Setup

We evaluate our approach on seven single-tree datasets: Iris, Wine, and Cancer from standard UCI/scikit-learn benchmarks [12], Breast, Heart, and Steel from SortingHat [6], and Website Phishing from a locally prepared three-class dataset export. The first three trees are trained with `scikit-learn`, the next three use pre-trained SortingHat trees, and Website Phishing uses a local exported tree retained for the final experimental section.

We compare three inference modes: *Hard clear*, which uses exact cleartext comparisons; *Adaptive clear*, which replaces hard comparisons with node-wise `SOFTSTEP` polynomial approximations; and *Adaptive HE*, which evaluates the same adaptive pipeline under CKKS. This separation allows us to measure the approximation loss Δ_{approx} , the additional CKKS-induced loss Δ_{CKKS} , and the average encrypted inference time per sample.

Experiments run in a CPU-only Ubuntu 22.04 Podman container. The HE layer uses Microsoft SEAL v4.1.2; the inference engine is implemented in C++17 with CMake, and data preparation uses Python/`scikit-learn`. We use three CKKS runtime profiles according to the feature dimension: *small* for $d \leq 4$, *medium* for $5 \leq d \leq 16$, and *large* for $d > 16$. Samples are packed with vertical SIMD packing: each sample occupies $d \cdot n_{\max}$ contiguous slots and multiple samples are batched in one ciphertext, with a maximum packed batch size of 32.

To support reproducibility, the source code of the proof-of-concept implementation is made publicly available online.³

6.2 Benchmark Scope

In addition to our implementation, we consider an Akavia-style SEAL pipeline, SortingHat, and Concrete-ML as adjacent baselines. These comparisons are contextual rather than controlled: the frameworks differ in model source, backend, sample count, batching, hardware assumptions, and execution mode. Our POC and the Akavia-style baseline use real SEAL execution, whereas the available Concrete-ML runs use FHE simulation; SortingHat relies on externally supplied pre-trained trees. Consequently, the main precision conclusions are drawn from our own hard-clear, adaptive-clear, and adaptive-HE modes, which share a common protocol.

³ Project repository

Table 4. Accuracy (%) across the retained inference modes. Timing is average adaptive HE inference time per sample. n denotes the number of evaluated samples.

Dataset	d	n	Hard	Ad. clear	Ad. HE	ms/samp.
Iris	4	20	100.0	95.0	95.0	286
Cancer	30	32	100.0	100.0	100.0	151
Wine	13	26	100.0	96.2	96.2	224
Breast	30	32	90.6	90.6	90.6	320
Heart	13	26	84.6	84.6	84.6	91
Steel	33	32	100.0	100.0	100.0	95
Website Phishing	9	32	81.25	75.0	75.0	239

Table 5. Decomposed adaptive accuracy loss.

Dataset	min_gap	Regime	Δ_{approx}	Δ_{CKKS}
Iris	0.017	II	5.0	0.0
Cancer	0.002	III	0.0	0.0
Wine	0.003	III	3.8	0.0
Breast	< 0.02	III	0.0	0.0
Heart	< 0.03	III	0.0	0.0
Steel	< 0.02	III	0.0	0.0
Website Phishing	0.005	III	6.25	0.0

6.3 Main Results and Precision Decomposition

Table 4 reports the main single-tree accuracy and runtime results for the three retained inference modes: hard cleartext inference, adaptive cleartext inference, and adaptive encrypted inference. Table 5 reports the two loss components defined in Eqs. (16) and (17). Table 6 then adds the retained random-forest runs for Road RF and Website Phishing RF.

The adaptive configuration is the main reference point. On Cancer, Breast, Heart, and Steel, adaptive HE matches the hard cleartext baseline, indicating that neither the polynomial approximation nor the CKKS evaluation changes the final prediction on these datasets. On Wine, Iris, and Website Phishing, the loss is already visible in adaptive cleartext and remains unchanged under CKKS. This shows that the observed degradation is caused by polynomial approximation rather than by encrypted evaluation.

These results support the precision-regime analysis of Section 5.3. Wine, Iris, and Website Phishing illustrate approximation-dominated cases, where replacing hard comparisons with polynomial soft indicators changes some decisions close to tree thresholds. By contrast, Cancer, Breast, Heart, and Steel remain stable under the adaptive pipeline, and the equality between adaptive cleartext and adaptive HE shows that the residual CKKS noise does not introduce additional measurable accuracy loss in this experimental campaign.

On the retained Website Phishing run, the drop from 81.25% hard cleartext accuracy to 75.0% adaptive cleartext and adaptive HE accuracy corresponds

Table 6. Retained random-forest results in the experimental section. n denotes the number of evaluated samples.

Dataset	Trees	n	Hard	Ad. clear	Ad. HE	/ ms
Road RF	5	32	100.0	100.0	100.0	/ 6061
Website Phishing RF	5	32	84.38	84.38	84.38	/ 1427

to two additional misclassifications over 32 evaluated samples, with no extra CKKS-only degradation.

The random-forest extension shows stable behaviour on the retained Road and Website Phishing runs. Road RF preserves the retained baseline, but its 32-sample batch is class-degenerate and is therefore treated as an implementation check rather than a balanced accuracy claim. Website Phishing RF remains at 84.38% in hard cleartext, adaptive cleartext, and adaptive HE. These results provide an initial scalability check showing that the retained forest configurations can preserve prediction accuracy under encrypted evaluation, while a broader evaluation on larger and more diverse forests remains future work.

6.4 Cross-Project Comparison

Tables 7 and 8 provide contextual comparisons with the available Akavia-style, SortingHat, and Concrete-ML traces. Because backends, models, sample counts, and execution modes differ, the values should not be interpreted as controlled evidence of algorithmic superiority. On the shared standard datasets, our retained traces show higher accuracy and lower latency than the Akavia-style SEAL baseline. For Website Phishing, our POC reaches 75.0% accuracy with 239.2 ms/sample, compared with 84.375% and 329.198 ms/sample for SortingHat, and 85.98% and 168855.77 ms/sample for Concrete-ML. Thus, the retained Website Phishing trace illustrates a precision–latency trade-off rather than a uniform advantage: SortingHat and Concrete-ML achieve higher accuracy, whereas our POC has lower reported latency. Concrete-ML remains a simulation-based reference and should not be interpreted as a directly comparable real-HE timing baseline.

Table 7. Cross-project accuracy comparison (%). “–” indicates that no value is available for that pair.

Dataset	Ours	Akavia	SortingHat	Concrete-ML
Cancer	100.0	78.1	–	93.3
Wine	96.2	61.5	–	100.0
Iris	95.0	90.0	–	100.0
Breast	90.6	–	90.6	100.0
Heart	84.6	–	81.8	81.8
Steel	100.0	–	100.0	100.0
Website Phishing	75.0	–	84.375	85.98

Table 8. Cross-project average HE inference time per sample (ms). Batch size is shown in parentheses when available.

Dataset	Ours	Akavia	SortingHat	Concrete-ML
Breast	320.3 (32)	–	154.6	190.5
Cancer	151.1 (32)	18996.3	–	430.5
Heart	90.5 (26)	–	35.1	537.1
Iris	286.1 (20)	12695.2 (10)	–	561.3 (10)
Steel	95.4 (32)	–	38.8	418.2
Wine	224.5 (26)	13274.6	–	417.7
Website Phishing	239.2 (32)	–	329.198	168855.77

6.5 Runtime Discussion

For single-tree adaptive HE, average latency ranges from about 90 ms/sample on Heart to 320 ms/sample on Breast, with Website Phishing at 239.2 ms/sample. On Website Phishing, the retained SortingHat and Concrete-ML traces report 329.198 ms/sample and 168855.77 ms/sample, respectively, although the latter corresponds to the available Concrete-ML simulation setting and is not directly comparable to real CKKS execution. Iris remains relatively costly because its smaller batch amortises rotations and key switching less effectively. RF latency is higher: about 1427 ms/sample for Website Phishing and 6061 ms/sample for Road. These results target offline or non-real-time use; lower-latency deployment requires further optimisation.

7 Conclusion

This paper studied privacy-preserving decision-tree inference under CKKS using HBDT-SUMPATH traversal and SOFTSTEP polynomial comparisons. We introduced a Soft-Adaptive degree-assignment strategy that adapts the polynomial degree to the local difficulty of each node, together with a precision-impact analysis that separates polynomial-approximation loss from additional CKKS-induced

loss. Across the retained single-tree experiments, adaptive HE matches Soft-Adaptive cleartext inference, showing no additional measurable CKKS-induced accuracy degradation under the evaluated settings. The observed losses on Iris, Wine, and Website Phishing are already present in cleartext and are therefore attributed to polynomial approximation. The random-forest experiments on Road and Website Phishing further show that the approach extends to ensembles, although at a higher computational cost.

Future work will investigate larger tree ensembles and gradient-boosted models such as XGBoost, with emphasis on ciphertext packing, tree-level parallelism, and encrypted score aggregation. We also plan to extend the framework from private inference to privacy-preserving training directly over encrypted data, which raises additional challenges for split selection, encrypted comparisons, gradient or statistic aggregation, and training-time efficiency. Furthermore, we will strengthen the current semi-honest security model toward malicious security and verifiable private inference, for example through zero-knowledge or other verifiable-computation mechanisms. Additional directions include stronger model privacy, min-gap-aware training, improved calibration, and adaptive CKKS parameter selection for larger and deeper tree-based models.

References

1. Akavia, A., Leibovich, N., Resheff, Y.S., Ron, R., Shahar, M., Vald, M.: Privacy-preserving decision trees training and prediction. *ACM Transactions on Privacy and Security* 25(3), 1–30 (2022)
2. Akhavan Mahdavi, R., Ni, H., Linkov, D., Kerschbaum, F.: Level up: Private non-interactive decision tree evaluation using levelled homomorphic encryption. In: *Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security*, pp. 2945–2958. ACM (2023). <https://doi.org/10.1145/3576915.3623095>
3. Azogagh, S., Delfour, V., Gambs, S., Killijian, M.O.: PROBONITE: Private one-branch-only non-interactive decision tree evaluation. In: *Proceedings of the 10th Workshop on Encrypted Computing and Applied Homomorphic Cryptography*, pp. 23–33. ACM (2022). <https://doi.org/10.1145/3560827.3563377>
4. Bhardwaj, D., Saravanan, S., Chandran, N., Gupta, D.: Securely training decision trees efficiently. In: *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, pp. 4673–4687 (2024)
5. Cheon, J.H., Kim, A., Kim, M., Song, Y.: Homomorphic encryption for arithmetic of approximate numbers. In: Takagi, T., Peyrin, T. (eds.) *ASIACRYPT 2017*, LNCS, vol. 10624, pp. 409–437. Springer (2017)
6. Cong, K., Das, D., Park, J., Pereira, H.V.L.: SortingHat: Efficient private decision tree evaluation via homomorphic encryption and transciphering. In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*, pp. 1647–1660. ACM (2022).
7. Cong, K., Kang, J., Nicolas, G., Park, J.: Faster private decision tree evaluation for batched input from homomorphic encryption. In: *International Conference on Security and Cryptography for Networks*. Springer (2024)

8. Dathathri, R., Saarikivi, O., Chen, H., Laine, K., Lauter, K., Maleki, S., Musuvathi, M., Mytkowicz, T.: CHET: An optimizing compiler for fully-homomorphic neural-network inferencing. In: *Proceedings of PLDI 2019*, pp. 142–156. ACM (2019)
9. Feng, Z., Xiong, H., Song, C., Yang, S., Zhao, B., Wang, L., Chen, Z., Yang, S., Liu, L., Huan, J.: SecureGBM: Secure multi-party gradient boosting. In: *IEEE International Conference on Big Data*, pp. 1312–1321. IEEE (2019)
10. Frery, J., Stoian, A., Bredehoft, R., Montero, L., Kherfallah, C., Chevallier-Mames, B., Meyre, A.: Privacy-preserving tree-based inference with TFHE. In: *International Conference on Mobile, Secure, and Programmable Networking*. Springer (2023)
11. Han, K., Ki, D.: Better bootstrapping for approximate homomorphic encryption. In: *CT-RSA 2020*, LNCS, vol. 12006, pp. 364–390. Springer (2020)
12. Kelly, M., Longjohn, R., Nottingham, K.: The UCI Machine Learning Repository. <https://archive.ics.uci.edu> (2024)
13. Kiss, A., Naderpour, M., Liu, J., Asokan, N., Schneider, T.: SoK: Modular and efficient private decision tree evaluation. *Proceedings on Privacy Enhancing Technologies* 2019(2), 187–208 (2019)
14. Ko, R., Akhavan Mahdavi, R., Yoon, B., Onizuka, M., Kerschbaum, F.: Silent-Wood: Private inference over gradient-boosting decision forests (2024)
15. Liang, H., Lu, H., Wang, G.: BPDTE: Batch private decision tree evaluation via amortized efficient private comparison. Cryptology ePrint Archive, Paper 2024/619 (2024)
16. Lin, G., Han, W., Ruan, W., Zhou, R., Song, L., Li, B., Shao, Y.: ENTS: An efficient three-party training framework for decision trees by communication optimization. In: *Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security*, pp. 4376–4390 (2024)
17. Lu, W., Zhou, J.J., Sakuma, J.: Non-interactive and output expressive private comparison from homomorphic encryption. In: *Proceedings of ACM ASIA CCS*. ACM (2018)
18. Paul, J., Tan, B.H.M., Veeravalli, B., Aung, K.M.M.: Non-interactive decision trees and applications with multi-bit TFHE. *Algorithms* 15(9), 333 (2022)
19. Sarpatwar, K., Nandakumar, K., Ratha, N., Rayfield, J., Shanmugam, K., Pankanti, S., Vaculin, R.: Efficient encrypted inference on ensembles of decision trees (2021)
20. Shin, H., Choi, J., Lee, D., Kim, K., Lee, Y.: Fully homomorphic training and inference on binary decision tree and random forest. In: *European Symposium on Research in Computer Security*, pp. 217–237. Springer (2024)
21. Tueno, A., Boev, Y., Kerschbaum, F.: Non-interactive private decision tree evaluation. In: *Data and Applications Security and Privacy XXXIV*, LNCS. Springer (2020)
22. Tueno, A., Kerschbaum, F., Katzenbeisser, S.: Private evaluation of decision trees using sublinear cost. *Proceedings on Privacy Enhancing Technologies* 2019(1), 266–286 (2019)
23. Xiao, X., Mi, G., Li, W., Huang, J.: Privacy-preserved approximate classification based on homomorphic encryption. *Mobile Networks and Applications* 24(6), 1905–1914 (2019)
24. Yuan, S., Li, H., Qian, X., Hao, M., Zhai, Y., Xu, G.: Efficient and privacy-preserving outsourcing of gradient boosting decision tree inference. *IEEE Transactions on Services Computing* (2024)

On Private Knowledge Retrieval and Private Inference for secure multi-agent systems

Vicenç Torra¹[0000–0002–0368–8037] and Maria
Bras-Amorós²[0000–0002–3481–004X]

¹ Department of Computing Science, Umeå University, Umeå, Sweden.

² Departament of Mathematics, Universitat Politècnica de Catalunya (UPC),
Barcelona, Catalonia, Spain.

E-mail addresses: `vtorra@cs.umu.se`, `maria.bras@upc.edu`

Abstract. Multi-agent systems provide a convenient framework for the massive deployment of agents based on Large Language Models (LLMs). It provides a sound framework where agents can interact with existing software to perform actions in an autonomous way. Agentic AI is the new term to denote these AI-based autonomous systems.

In real multi-agent systems, agents need to deal with malicious agents some of which can exploit security vulnerabilities to do harm. A memory poisoning attack is an example of exploiting a vulnerability to modify agents knowledge and, thus, affect its behavior. In an agent, there are different types of memory systems (semantic, episodic, and short-term) which have their own characteristics and for which different memory poisoning attacks may be developed.

In this paper we focus on semantic memory, we discuss its attack, and mitigation strategies. We introduce private knowledge retrieval and private inference. We propose a signature system in the knowledge base to prevent from undesired manipulation. This should avoid memory poisoning attacks by introducing security by design in the construction of the semantic memory. In our architecture, an agent contains a local Prolog-like inference system (i.e., using Horn clauses to represent knowledge) which accesses an external knowledge base to get the required knowledge. This externally signed knowledge base is accessed privately. We provide different implementations in Python using different privacy levels including k -anonymity and information-theoretical privacy (IT-PIR).

Keywords: Agentic AI · Memory poisoning attacks · Short-term memory

1 Introduction

Agentic AI is becoming the new methodology to develop distributed AI-based systems taking advantage of Large Language Models (LLMs). Agentic AI relates with the well-established field of Multi-Agent Systems (MAS), where autonomous agents collaborate in solving problems together. That is, instead of an agent solving a problem alone accessing other types of software tools, a myriad

of agents collaborate and coordinate to solve problems. See discussions in [12] and [5] on the connections between both fields. Multi-agent systems and agentic AI allow us to build sophisticated software architectures. An example is the AI-orchestrated cyber espionage campaign described by Anthropic [2] based on Claude.

When implementing and deploying multi-agent systems, we need to take into account that there may be malicious agents that can exploit both security vulnerabilities and design principles so that they can affect other agents behaviors to do harm. It is also important to note that interactions between agents are not one-shot. During the life-cycle of an agent, malicious agents can have long interactions that can be exploited for long-term attacks.

Memory poisoning attacks are an important type of attack. They consist on a non-authorized modification of agents memory to modify its behavior. Memory is a key component in any agent, as it stores interactions, experiences, and knowledge. Then, naturally, memory affects how agent’s decisions are made and actions implemented. Memory poisoning, and its relation with LLMs and LLMs-based agents, has a relationship with prompt hacking [11] (on both prompt injection and jailbreaking) with the goal to cause unintended agent’s behavior. Nevertheless, in prompt hacking the goal is to build prompts that cause a certain (malfunctioning or unintended) behavior of the LLM-based agent. There is no intended access to memory.

It is important to underline that memory is not a monolithic structure. Agents have different memory systems, each of them with their own characteristics, and memory poisoning attacks can be designed and implemented for each of them. The three main types of memory systems are short-term, episodic, and semantic memory. Their difference is in their duration and consolidation. I.e., from recent interactions to long term and consolidated knowledge.

Contributions. In this paper we focus on semantic memory and discuss and propose mitigation strategies for semantic memory poisoning attacks.

We first argue that the formalization of memory poisoning as it is proposed in the recent paper by Chen et al. [7] is not always meaningful. They formalize poisoning in a way similar to data and model poisoning, which has been studied for several years in e.g. the federated learning community. Such approach, based on defining an optimization problem, looks for optimal modifications to produce a certain agent’s behavior. We challenge their assumptions and discuss that there are security solutions that help us to mitigate these attacks.

Second, we introduce private knowledge retrieval and private inference for secure multi-agent systems as a way to mitigate semantic memory poisoning attacks. Our threat model consists of malicious agents that want to poison the memory of another agent. We want to protect semantic memory (both knowledge and inferences) by means of preventing malicious agents to know what is stored in memory, and make tampering difficult. Our approach is to keep the knowledge base external but having logical inferences local. We develop a proof of concept, based on private information retrieval, which permits us to store knowledge in terms of Horn clauses, as in Prolog, and provide logical inference. Agents

to implement local logical inferences can request external knowledge, which is signed to avoid tampering, and supplied privately to the agent.

Our implementation demonstrates the technical solutions that can mitigate memory poisoning. We have also implemented both strong and light solutions. E.g., we have implemented both IT-PIR and k -anonymity. Digitally signed knowledge makes tampering more difficult; private knowledge retrieval reduces the risk of unauthorized access to agent’s knowledge and inference processes. We evaluate our approach experimentally and discuss computational costs.

Paper structure. The structure of the paper is as follows. In Section 2, we describe memory systems and memory poisoning attacks in multi-agent systems. Then, in Section 3, we introduce our private inference system, and, in Section 4, we describe our experiments and present our analysis. The paper concludes with a discussion and a description of some research directions.

2 Memory systems and their poisoning attacks in MAS

Agentic AI needs a memory to store available knowledge, partial results and interactions with other agents. Memory, however, is not a single and compact structure but it is often a composite element including different memory systems. This discussion on memory is agnostic on the type of agents we are considering. It is valid for both LLMs-based agents as well as for more classical architectures [35] as deliberative agents based on BDI-architectures (Belief-Desire-Intention architectures) that may include knowledge in terms of logical expressions.

2.1 Memory systems

Existing literature [8,20,21,23,36] classifies different memory systems in different ways. In this work, following recent trends, we distinguish three types of memory systems based on its duration and consolidation. We classify them from short-term to long-term memory. Next we describe these three types of memory systems from our specific focus on symbolic semantic memory.

- **Short-term memory.** This type of memory stores the current conversation and keeps track of existing interactions with other agents. Partial results as well as partial inferences from existing knowledge can also be included in short-term memory.
- **Episodic memory.** This type of memory stores past interactions and other not-so-recent information. It is often a consolidation of short-term memory. That is, summaries of previous short-term memories.
- **Semantic memory.** This type of memory system corresponds to domain-specific knowledge or factual knowledge. It is typically a relatively permanent structure with very low frequency updates. A typical example corresponds to the capitals of countries (e.g., Paris is the capital of France), and established medical knowledge about certain illnesses. There are different ways to implement semantic knowledge. A common way is to use a symbolic approach [3,16]. In this case, knowledge is typically represented by means of

logical expressions, ontologies, or Bayesian models (graphical models). In this case, a knowledge base that contains domain-specific knowledge is combined with an inference mechanism that allows the agent to infer about certain facts (its desired queries) from already known information (facts in the knowledge base or facts in short-term memory) and previously established knowledge. Resolution and Prolog-like inference is the standard mechanism for knowledge expressed by logical expressions. An alternative is to use Retrieval-Augmented Generation (RAG) which permits us to incorporate specific knowledge into LLMs-based agents. Some references use the term *declarative memory* [20,21] which can be seen as a synonym of *semantic memory*. We may also consider *procedural memory* as a type of semantic network. It corresponds to implicit knowledge to perform an action. It is implicit in the sense that it is not possible to reason about how the action is done.

2.2 Semantic memory poisoning attacks

A memory poisoning attack consists of an undesired modification of the agent’s memory by another agent. The goal of memory poisoning is to modify on purpose an agent’s behavior. Literature on memory poisoning is about studying possible attacks as well as developing mitigation strategies. These attacks and strategies typically follow the machine learning perspective, and are similar to existing attacks and mitigation strategies for data and model poisoning in federated learning. In this section, we describe the typical attack and object later about its relevance. As we discuss in more detail in the rest of the section, agents secure by design can overcome some of these problems.

For semantic memory, the literature considers attacks following the seminal paper by Chen et al. in NeurIPS [7]. In this attack, an agent computes the optimal modification of the semantic memory in order to affect agent’s behavior. According to the authors [7] the “assumption aligns with practical scenarios where the memory unit of a victim agent is hosted by a third-party retrieval service or directly leverages an unverified knowledge base”.

The attack model considers a poisoned database $\mathcal{D}_{poison}$ with a trigger x_t defined in terms of the clean database $\mathcal{D}_{clean}$ to which an attacker injects adversarial key-value pairs. These injected key-value pairs are denoted by $\mathcal{A}(x_t)$ where $\mathcal{A}(x_t) = \{(k_1(x_t), v_1), \dots, (k_{|\mathcal{A}(x_t)|}(x_t), v_{|\mathcal{A}(x_t)|})\}$. Then,

$$\mathcal{D}_{poison}(x_t) = \mathcal{D}_{clean} \cup \mathcal{A}(x_t). \quad (1)$$

Let us denote by $\mathcal{E}(q, \mathcal{D})$ the inference of the semantic memory given query q and knowledge or database $\mathcal{D}$. Then, the output of a query q from the poisoned database is denoted by $\mathcal{E}(q, \mathcal{D}_{poison}(x_t))$, and, similarly, adding the trigger x_t to a query q the output will be $\mathcal{E}(q \oplus x_t, \mathcal{D}_{poison}(x_t))$. Attacks are to cause a modification of the output under the presence of x_t so that instead of producing a benign output a_b we have a malign output a_m leading to a malign action. Examples of malign actions are removing the whole home directory or buying a wrong flight ticket.

Under these assumptions, an attack is formalized as a multi-objective optimization problem consisting of the following two objectives:

- Maximization of adversarial output. If a_m is the target malicious action, the attacker wants to maximize this action for triggers. That is, given a sample distribution of input queries π , the retrieval of q (where q is drawn from a query distribution $q \sim \pi$) poisoned with x_t (i.e., $q \oplus x_t$) produces a_m with high probability. Formally,

$$\text{Maximize } \mathbb{E}_{q \sim \pi} [\mathbb{1}(LLM(q \oplus x_t, \mathcal{E}(q \oplus x_t, \mathcal{D}_{poison}(x_t))) = a_m)], \quad (2)$$

where $\mathbb{1}(\cdot)$ denotes the indicator function. That is, a function that is one when its argument is true, and zero otherwise.

- Minimization of the effect on non-targeted queries. That is, the output should not be affected and the action produced by the LLM-based agent should be the benign a_b . Formally, with high probability it produces a_b and not a_m ,

$$\text{Maximize } \mathbb{E}_{q \sim \pi} [\mathbb{1}(LLM(q, \mathcal{E}(q, \mathcal{D}_{poison}(x_t))) = a_b)]. \quad (3)$$

This model assumes that the clean database is known (see Equation 1), updates are possible (i.e., that we can add these $\mathcal{A}(x_t)$ key-value pairs), and we know local-agents’ LLM-s outputs (see $\mathcal{E}$ in Equations 2 and 3). Mitigation strategies need to address these key aspects. Our mitigation strategies directly challenge these assumptions.

2.3 Mitigation strategies for semantic memory poisoning attacks

As we have explained above, semantic memory is about established knowledge that is mainly static, and updates are only needed from time-to-time. We propose the following three strategies to avoid semantic memory poisoning: (i) secure memory mechanisms to avoid internal memory poisoning attacks, (ii) secure memory updating algorithms for knowledge bases and (iii) secure communications with external agents as well as communication logs. Each of these requirements is discussed below.

The semantic memory attack model assumes that the memory is external to the agent and hosted by a third party. We discuss this fact, and we consider below both the case of internal and external memory.

(i) Secure memory mechanisms. Internal secure memory can be implemented using hashing and signatures. Hashing allows to detect any undesired changes in memory, and signatures allow to ensure provenance of the memory components. Moreover, cryptographic protocols can help us to make local memory private, and thus, make $\mathcal{D}_{clean}$ unknown to attackers, which prevents the attacks associated to Equations 1, 2 and 3.

Same tools (hashing and signatures) apply when the semantic memory “is hosted by a third-party retrieval service”. This would limit malicious manipulation of the memory. In short, ensuring safe memory in trusted third parties is

similar to the case of avoiding corrupted databases in the cloud. This approach permits intruders to study the optimization problem above as $\mathcal{D}_{clean}$ may be accessed, but intruders will not be able to manipulate the knowledge base. We have implemented this approach in our system. In our implementation, each knowledge item (including its provenance information, see below) is signed to avoid its manipulation.

In the case of external untrusted knowledge based servers providing inference, there are solutions for deep learning models. They are based [25] on Fully Homomorphic Encryption (FHE). These solutions are costly and only available in practice for small neural networks. For other types of data-driven models there are some FHE solutions. See e.g. the case of decision trees. Up to our knowledge, there are no private-inference models in the literature for knowledge base inference as the one discussed here below. They would be costly in any case, and more costly than decision trees.

Our alternative, discussed later in detail, is to keep the static part of the semantic memory mechanism external and signed, while maintaining the inference system protected and local to the agent. Formally, with respect to Equations 2 and 3, this requires $\mathcal{E}(\cdot)$ to be local and protected, which prevents malicious agents from computing $\mathcal{E}(q \oplus x_t, \mathcal{D}_{poison}(x_t))$. Note that although our inference engine is based on Prolog, the knowledge and knowledge-base servers are agnostic to the inference engine (i.e., other forms of logical inference could also be implemented). Moreover, digital signatures prevent unauthorized updates to $\mathcal{D}_{clean}$, thus preventing Equation 1 from applying. In other words, a malicious server cannot modify the knowledge base without being detected, and local inference is therefore performed over authenticated knowledge. This way inference is expected to be reasonably correct. We introduce private knowledge retrieval and private inference according to this design in Section 3.

(ii) Secure updating of knowledge bases. Knowledge bases need to be updated from time to time. Provenance structures can ensure integrity for data but also for the provenance structures themselves so that no agent can forge provenance data. Note that data provenance can ensure completeness (i.e., that all actions that are relevant to computation are detected and represented). Provenance structures [6,33] can provide both coarse-grained and fine-grained information about updates and modifications of knowledge.

We have included basic provenance information in our knowledge base. Each knowledge item (i.e., Prolog-like Horn clauses, rules, and facts, as explained in more detail below) includes metadata specifying its source. Moreover, each knowledge item, together with its provenance information, is digitally signed by the provenance agent using its private key and a SHA-256-based signature scheme. This allows agents to verify the integrity of the received information before inference using the agent’s public key.

(iii) Secure communication and communication logs. Communication with trusted agents should be secured so that information and knowledge are

transmitted safely. Public-key encryption and digital signatures will ensure safe transmission. However, our current system implements neither secure communication mechanisms nor communication logs. This limitation is less relevant in our prototype as we do not support agents running in parallel, and everything is executed on a single machine.

3 Private inference in tipp++

In this section we describe our architecture for private knowledge retrieval and private inference as well as its components. In short, the architecture consists of a set of agents deployed to do a task (or a set of tasks). It also includes agents providing knowledge as a service. That is, there are knowledge base servers that provide knowledge on demand. Agents include local knowledge base inference mechanisms that may require access to this external knowledge.

We begin describing the Prolog-like mechanism which is implemented locally within the agent, then the knowledge base client which is internal to the agent and that can access the external knowledge base when local knowledge does not suffice, and finally, we describe the implementation of the external database. Our software is published online [37].

3.1 Prolog-like inference mechanism

We have implemented a Prolog-like inference mechanism in Python where knowledge is represented in terms of Horn clauses (see Figure 1). Thus, we can represent both facts and rules (implications). Recursion is allowed in rules. A difference with Prolog is that disjunction is not implemented in our rules and only pure Horn clauses can be used. Recall that from a logical perspective, this is not a constraint on the logical expressiveness of the Horn clauses because a clause with disjunction (e.g., $p:- q \vee r$) is logically equivalent to a set of Horn clauses with only conjunction (e.g., $p:- q$ and $p:- r$).

Predicates follow the syntax in Prolog, with the name of the functor and number of arguments. We use the same convention used in Prolog, strings starting with upper-case letters represent variables and those starting with lower-case letters represent constants. The number of arguments of a predicate corresponds to its arity. We do not implement the wildcard variable “_” commonly used in Prolog. Note that not implementing “_” does not limit the expressiveness of our knowledge representation formalism. For example, in Prolog the fact `member(X, [X|_])` can be implemented as well with `member(X, [X|YY])`.

In addition to constants (and variables) we have also implemented lists. They are defined in terms of their constructors `nil` and `cons`. Recall that `nil` defines the empty list, and `cons` adds an element to an already existing list (i.e., `nil` corresponds to generating `[]`, `cons(a,nil)` corresponds to `[a]`, and `cons(a,[b,c,d])` corresponds to `[a,b,c,d]`). We have implemented a preprocessing method so that the knowledge base can be defined using lists in the

mortal/1	$\Rightarrow [\forall x, \text{man}(x) \Rightarrow \text{mortal}(x),$
	$\forall x, \text{woman}(x) \Rightarrow \text{mortal}(x)]$
woman/1	$\Rightarrow [\text{woman}(\text{hypathia})]$
man/1	$\Rightarrow [\text{man}(\text{socrates})]$
vehicle/1	$\Rightarrow [\forall x, \text{car}(x) \Rightarrow \text{vehicle}(x)$
	$\forall x, \text{bicycle}(x) \Rightarrow \text{vehicle}(x)]$

Fig. 1. Key-value pairs for the knowledge-base considering only Prolog-like inference systems and queries. Key values correspond to predicate names and arity.

standard way (i.e., using expressions as $[a, b, c]$) and then internally translated into lists using the standard constructors `nil` and `cons`.

The inference mechanism is implemented following standard Prolog resolution. A query, or in logical terms a request to prove or refute a logical expression, can be a single predicate possibly with uninstantiated variables or a sequence of predicates also possibly with uninstantiated variables. In other words, a sequence of one or more subgoals. Then, the engine selects the first goal and either satisfies it with a fact (via appropriate unification of variables), or expands this goal using a rule/implication. In the latter case the subgoal is replaced by a list of other subgoal(s) and thus, possibly, enlarging the sequence of goals. The process is repeated until all goals are satisfied. When the use of a certain rule leads to a dead end (the sequence of goals cannot be fulfilled) the process backtracks and another rule is selected. If there is no way to fulfill the original goals because there are no further rules to apply, the inference fails. Using the standard closed-world assumption (CWA) in Prolog (i.e., standard Prolog semantics), this means that the query is false.

Our implementation differs with respect to standard Prolog because by default we provide all possible instantiations of a predicate up to a given recursion depth. For example, the output of a proof of `mortal(X)` would be a list of elements `X` that are mortal. In our implementation this includes `socrates`, `aristotle`, and `hypathia`. We use this approach for two reasons, (i) we do not implement the cut operator (i.e., the `!` in Prolog), and (ii) we do not require facts and rules to be defined in a particular order. In Prolog, different orderings can produce solutions in different order, and can even cause the program to loop infinitely.

3.2 Knowledge base client

Our knowledge base inference system is run locally by the agent. That is, agents have their own knowledge base client which can contain local knowledge as well as provide access to the knowledge base server to obtain external knowledge.

When the local inference system needs to resolve a query, it queries the local knowledge base using the pair (predicate name, arity). The knowledge base client

will provide the corresponding Horn clauses to the inference system. Figure 1 illustrates key-value pairs indexed with the pairs (predicate name, arity) and returning Horn clauses. If the knowledge-base client does not have any available Horn clause for this pair, or if the relevant knowledge has expired (each knowledge item is assigned a time-to-live, or TTL), the query is forwarded to the knowledge-base server. Results of external queries are then processed and appropriately sent to the inference system.

In addition, the knowledge retrieved will be cached by the knowledge base client, so that there is no need to query external servers again for the same knowledge. Our main reason for applying caching is that solving a predicate usually requires multiple access to the same facts, and because rules are recursive. So, external access will increase computation time. Moreover, note that repeated access to the same Horn clauses can give hints to intruders about which clauses are being queried and can compromise privacy as well as give hints for knowledge poisoning.

For example, when the agent queries `append([a], L, [a,b])` to the local inference system, it will try to retrieve all Horn clauses with signature “append/3” from the knowledge base client. If no clause (no facts or rules) are found in the local knowledge base, the query is forwarded to the external knowledge base server(s). For this example, our knowledge base will return the following:

```
("append",3): [
  {"head": ["append","nil","X","X"], "body": [],
   "meta": {"predicates":"","source":"db3"}},
  {"head": ["append",("cons","X","Xs"), "Ys", ("cons","X","Zs")],
   "body": [["append","Xs","Ys","Zs"]],
   "meta": {"predicates":"other", "source":"db3"}},
```

Additional possible improvements. We implement retrieval solely based on predicate name and arity. It is possible to query taking into account the arguments as well. This will reduce the amount of clauses transferred from knowledge-base servers to knowledge-base clients. This approach is used by Just In Time Indexing (JITI) utilities in SWI-Prolog [38]. Another optimization is tabling (a memoization technique). Tabling consists of storing the results of predicate calls. In other words, it is about caching and reusing answers. None of these optimization techniques have been implemented in our system.

3.3 Knowledge base server

In our threat model, knowledge-base servers are not trusted, agents seek to protect their requests, and knowledge items must be protected. Protection of agent’s requests is achieved through private knowledge retrieval, which is based on private information retrieval (PIR) and described in the next section. Knowledge items are protected together with their provenance information. Specifically, each knowledge item, together with its provenance information, is digitally signed by the provenance agent using its private key and a SHA-256-based signature scheme, as described in Section 2.3 (ii). In our implementation, all knowledge

base servers are initialized at system startup, and they are provisioned with all the required signed knowledge.

3.4 On the implementation of private knowledge retrieval

As we have just mentioned, private knowledge retrieval is based on private information retrieval. Chor et al. [9,10] proved that information-theoretical privacy can be achieved with the assumption that there are copies of the same database in at least two different queryable storage servers which do not communicate with each other. When this is the case, each server gets no information on the item retrieved by the user. In contrast, for a single server information-theoretic privacy can only be provided when the whole database is retrieved.

There are relaxations of this approach. For example, Kushilevitz and Ostrovsky [22] relaxed information-theoretical privacy to computational privacy and presented a method for constructing single-database PIR based on the Goldwasser-Micali public-key encryption scheme [15]. Many other single-database PIR schemes have appeared since then. See the survey [24], and more recently [18,4,34] and the survey [1].

In our implementation, we follow the approach of Chor et al. [9,10] to implement private knowledge retrieval using two copies of the database (see also [31]). For this, we need to index knowledge items. We have already specified above that the inference engine and the knowledge base clients request Horn clauses based on the pairs (predicate-name, arity). We use these pairs to index knowledge, and, therefore, we compute an index or hash from them. As the number of Horn clauses (facts and rules) and their lengths associated with a given index will be typically different, the database is padded so that all indices in the knowledge base have the same length.

To access a given index i , the agent client builds two random queries differing in the value associated to i . The agent submits one query to each knowledge base server, each server provides a string of bits that are xor-ed bits of the received query. These two responses are also xor-ed, and the resulting string is decoded providing the desired result, i.e., facts and rules associated to index i .

The agent's knowledge-base client receives the requested knowledge, verifies its integrity using the associated provenance information and digital signature, caches it provided the verification is successful, and forwards it to the inference engine for inference.

In addition to the approach that directly follows PIR, we have implemented two other alternative solutions that are more light-weighted. They are as follows.

- Single server, k -anonymity [27]. Given k , the agent submits the desired index i and $k - 1$ other random indices. The server provides all Horn clauses associated to the k indices, and the client selects the desired one.
- Single server, buckets. All knowledge (Horn clauses) in the knowledge base is divided into b buckets at the initialization stage. Then, for a desired index i , the agent determines in which bucket this knowledge is, and retrieves all associated knowledge (all Horn clauses) from the server. The agent then selects the desired one.

In what follows, we will use the following notation. We denote by n_{pa} the number of different pairs (predicate-name, arity) in the knowledge-base, by n_h the number of Horn clauses in the knowledge base, and by r the number of bits in the padded database.

In summary, we have implemented three alternative private knowledge retrieval protocols. Each implementation has different formal privacy and security guarantees, as well as different computational requirements. The stronger privacy guarantees with the, in principle, largest computational time corresponds to the implementation of non-colluding knowledge bases. In contrast, k -anonymity and buckets are less computationally demanding but at the same time providing less privacy and security guarantees. Among k -anonymity and buckets, its privacy guarantees and computation requirements depend on the parameter k and the number of buckets. So, if $b = n_{pa}/k$ then the privacy guarantees of the two approaches are similar, as a query will return a similar number of items. Note however that multiple access to the same index is deterministic for buckets but not for k -anonymity.

3.5 Untrusted agents and agent collusions

In our implementation we have considered two servers to provide privacy preserving knowledge retrieval. In addition, knowledge is digitally signed to prevent tampering. Moreover, multiple servers provide interesting mechanisms for guaranteeing robustness and reliability for information storage [29,30]. In this paradigm, encoding of the information stored by the servers (mainly by the use of maximum distance separable codes) prevents from the loss of information in case some servers get damaged or incommunicated and ensures reliability of the information, in case that a (small) number of servers send perturbed or directly fake information. This would provide additional security guarantees against memory poisoning attacks by some of the servers. However, when a number of servers is used, some of them may collude. Some solutions against this kind of collusions using error correcting codes can be found in [14,17]. In these references information retrieval is generalized to file downloading. Another alternative is peer-to-peer PIR [13] in which a user is cloaked in a peer-to-peer user community, where peers submit queries on behalf of other peers and conversely.

Access to knowledge bases and ontologies can also be implemented using these approaches. In this way, untrusted collusions of servers will not know which knowledge items are accessed, and malicious actions could be mitigated. That is, we could minimize compromising memory and future agent's actions. Locally repairable codes (LRC) [19,28], that are used for large databases in practice, can be used for this purpose.

4 Experiments and analysis

As described in the previous section, we have implemented a prototype of our architecture in Python. This prototype includes the local Prolog-like inference

	knowledge bases	sent communication	received communication	selection/ computation
k -anonymity	1 KB	$k \cdot 64$	$k \cdot r$	select 1 of k
b -buckets	1 KB	$1 \cdot 64$	$m \cdot r$	select 1 of b
2-PIR	2 KB	$2 \cdot n_{pa}$	dr	XOR($2r$)
2^d -PIR	d KB	$d \cdot n_{pa}^{1/d}$	dr	XOR(dr)

Table 1. Costs on bits for private knowledge retrieval implemented using k -anonymity, b -buckets, two non-colluding databases (2-PIR), and 2^d knowledge databases. 64 above represents the length in bits of integers (assuming long long int). Here, r is the number of bits associated to each index in the knowledge base, k the parameter associated to k -anonymity, b is the number of buckets and $m = n_{pa}/b$ is the number of rows retrieved by b -buckets.

mechanism. Although all components run on a single laptop, we simulate both the knowledge-base server(s) and the knowledge-base client by providing separate software implementations for each role.

We are assuming that the server is not trusted, nevertheless, we do not allow servers to collude. The external knowledge base is indexed taking into account the predicate name and its arity. Therefore, the size of the database in number of records depends on how many distinctive pairs (name, arity) exist. Then, we have implemented a set of queries and evaluated their execution time experimentally. In addition, we assess the complexity of querying the system. We begin with the analysis of the complexity of the private information retrieval approaches introduced in Section 3.4.

4.1 Complexity analysis

In the case of two non-colluding knowledge bases, each one with n_{pa} keys, the agent is required to send n_{pa} bits to each server (so, communication requires $2n_{pa}$ bits) and this agent will receive r bits from each server (so, communication requires $2r$ bits). Here r corresponds to the number of bits of a knowledge item of the padded database. Then, the agent performs a xor of these $2r$ bits obtaining a string of r bits. Our solution is not optimal. There are alternative solutions for 2^d databases for a given d in which the protocol sends $d \cdot n_{pa}^{1/d}$ bits in total and the agent receives from each database r bits (i.e., dr bits are received). Here the agent applies a xor of these $d \cdot r$ bits obtaining r bits.

In the case of k -anonymity, the intruder sends k indices, say integers of 64 bits (so, it sends $k \cdot 64$ bits) and receives k knowledge items. Considering as above that they are r bits long, this means that $k \cdot r$ bits are received. These strings need to be filtered to obtain the final r bits.

In the case of buckets, the approach has a cost analysis similar to the one of k -anonymity, except for the transmission from client to server, which is just the number associated to a set of buckets. So, say, 64 bits. Then, for a number of buckets equal to b we will have that each query will retrieve $m = n_{pa}/b$ indices

KB	Mean	St. dev.	Min	Max
baseline	0.17787843	0.00276265	0.17516215	0.18181177
bucket	0.16892560	0.00250258	0.16619691	0.17300868
Anon k4	0.17944365	0.00301458	0.17585521	0.18262578
Anon k6	0.17852928	0.00138278	0.17638855	0.17977013
Anon k8	0.18061575	0.00335783	0.17653908	0.18402709
Anon k10	0.17887916	0.00275545	0.17525422	0.18126626
it2pir	0.22309446	0.00500567	0.21711705	0.22866296

Table 2. Execution time statistics for different knowledge base access mechanisms, five runs, each accessing the knowledge bases as first time (i.e., case Non-cached).

from the knowledge base. So, the agent will receive $m \cdot r$ bits that need to be filtered to obtain the final r bits. Costs are summarized in Table 1.

4.2 Computational experiments

To test our system, we have defined a knowledge base consisting of $n_h = 99$ facts and rules, which corresponds to $n_{pa} = 46$ entries in the database. We have included a variety of Horn clauses, including both facts and rules, with non-recursive and recursive rules, and implementing as well two predicates that are benchmarks from SICStus Prolog [39]. In particular, we have implemented **nreverse** (naive reverse) which for a list of $n = 30$ elements requires a recursion of depth $(n + 1) + 2n(n + 1) = 31 + 465 = 496$, and **zebra**.

Then, we have considered a set of 18 queries and executed them in different configurations. We have also executed each configuration several times. The configurations we have considered are the following ones.

- “baseline” which means that a single external knowledge base is used, no privacy preserving access is applied,
- “pir_bucket” which means that we use a single external knowledge base implementing private information retrieval with buckets,
- “pir_kAnon” which means that we use a single external knowledge base implementing private information retrieval with k -anonymity, and
- “itpir_2servers” which means that we use the PIR protocol with two external knowledge bases.

Note that each of these types of executions are implemented for each of the above configurations. As queries in the external knowledge base are cached by the knowledge client, we have repeated executions in different ways. Computation times for each type of execution are provided.

- **Non-cached.** 5 executions of each query as if the query access was new each time but allowing caching for recursion as well as within the set of queries. For example, the first inference associated to **nreverse** requires an external knowledge base access to retrieve the associated Horn clauses,

KB	Mean	St. dev.	Min	Max
Baseline	0.0969	0.1119	0.0178	0.1761
Bucket	0.0919	0.1081	0.0154	0.1684
Anon k4	0.1006	0.1142	0.0198	0.1814
Anon k6	0.0976	0.1143	0.0168	0.1785
Anon k8	0.0978	0.1129	0.0180	0.1776
Anon k10	0.0987	0.1114	0.0200	0.1775
it2pir	0.1192	0.1503	0.0130	0.2255

Table 3. Execution time statistics for different knowledge base access mechanisms corresponding to two runs with cache. That is, first execution of the queries uses external access, and the second never does (case Standard caching).

but no external access is required for recursive computation of **nreverse**. Moreover, no external access is required for another call of **nreverse** within the same set of queries.

- **Standard caching.** Two executions of a query with cache. That is, we have executed each set of queries twice. Nevertheless, external access is only applied once, and all the other executions use cached Horn clauses. Because of that, even if the first execution required external access, the second execution never requires this external access, and, thus, the overall time for executing the set of queries takes much less time.
- **All cached.** Baseline with no external query. In order to assess the execution costs using the external knowledge base, we also provide mean execution times when no external knowledge base is used. These results are computed using only cached Horn clauses.

The execution times for the first set of experiments with the minimum number of cached Horn clauses (case Non-cached) are provided in Table 2. Then, execution times for the second set of experiments with partial caches (case Standard caching) are provided in Table 3. Finally, execution times for all Horn clauses cached (case All cached) are provided in Table 4. As expected, the longest execution time is for the case non-cached, Table 2.

The computational results displayed in these tables for the two PIR databases are surprisingly good. In the case non-cached, its time is 0.17s for baseline and 0.22s for it2pir. It seems that the implementation of the XOR on the knowledge base is extremely fast and comparable to the other expected-light-weight implementations that need to generate multiple queries, forward these queries, and then filter the results of these queries. Query generation and post-processing is implemented in Python.

Note that as all the query processing is performed by the knowledge base client and knowledge base server, execution times are in fact independent of the logical inference engine (our Prolog engine). In other words, the Prolog engine is agnostic on the type of knowledge base server used and it only queries the local knowledge base client and gets appropriate Horn clauses in return.

KB	Mean	St. dev.	Min	Max
Baseline	0.0135	0.0027	0.0117	0.0176
Bucket	0.0149	0.0038	0.0120	0.0205
Anon k4	0.0158	0.0044	0.0121	0.0217
Anon k6	0.0140	0.0016	0.0125	0.0161
Anon k8	0.0126	0.0016	0.0116	0.0150
Anon k10	0.0144	0.0021	0.0131	0.0175
it2pir	0.0129	0.0007	0.0125	0.0140

Table 4. Execution time statistics for different KB access mechanisms corresponding to five runs with no external access (i.e., baseline, case All cached).

5 Conclusions

In this paper we claim that the standard formalization of memory poisoning attacks for semantic memory can be mitigated using secure memory mechanisms, secure updating of knowledge bases, and secure communication and communication logs. We implemented a knowledge base which includes provenance and digital signatures to avoid knowledge tampering. We introduced a private knowledge retrieval and private inference as a way to provide secure inference. We described our implementation and provided experiments and analysis of its cost.

We mentioned in the paper that there are some ways to improve the performance of the inference system that were not considered and are left as possible extensions. In relation to knowledge retrieval, we considered retrieval by predicate name and arity. More sophisticated retrievals are possible taking into account the arguments of the predicates (e.g., as Just In Time Indexing, JITI, utilities in SWI-Prolog). Also, it would be possible to use tabling, memoization of inference results.

In our implementation we considered only a single external knowledge base, or two knowledge bases with replicated content. Our approach can be easily extended to multiple knowledge base servers with different knowledge. Specialized knowledge bases will offer their services listing their knowledge (i.e., providing appropriate hashing functions). In this case, agent’s client would proceed in a way analogous to our current system, but deciding first to which server it needs to send a request. I.e., instead of having a single hash function providing an index for the pair predicate name/arity we would have multiple hash functions, one for each knowledge base.

Acknowledgements

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. Support by Swedish Research Council (VR 2023-05531) is also acknowledged. The work was also partially supported by project PID2024-156636NB-C21 (MATSE) funded by MCIN/AEI/10.13039/501100011033/ FEDER, UE.

References

1. Alfarano, G.N., Khathuria, K., Weger, V, (2023) A survey on single server private information retrieval in a coding theory perspective, *Appl. Algebra Eng. Commun. Comput.*, vol. 34, no. 3, pp. 335-358.
2. Anthropic (2025) Disrupting the first reported AI-orchestrated cyber espionage campaign, [anthropic.com](https://www.anthropic.com).
3. Baura, D., Calvanese, D. (2025) Assessing Privacy Requirements for Controlled Query Evaluation in OBDA, *Proc. MDAI 2025* 183-197.
4. Bordage, S., Lavauzelle, J. (2021) On the privacy of a code-based single-server computational PIR scheme, *Cryptogr. Commun.*, vol. 13, no. 4, pp. 519-526.
5. Botti, V. (2025) Agentic AI and multiagentic: are we reinventing the wheel? [ArXiv:2506.01463](https://arxiv.org/abs/2506.01463).
6. Chapman, A., Lauro, L., Missier, P., Torlone, R. (2024) Supporting Better Insights of Data Science Pipelines with Fine-grained Provenance, *ACM Transactions on Database Systems* 49:2. Article No.: 6, pages 1 - 42.
7. Chen, Z., Xiang, Z., Xiao, C., Song, D, Li, B. (2024) AGENTPOISON: Red-teaming LLM Agents via Poisoning Memory or Knowledge Bases, *NeurIPS 2024*.
8. Chong, H.Q., Tan, A.H., Ng, G.W. (2007) Integrated cognitive architectures: a survey, *Artificial Intelligence Review* 28 103-130.
9. Chor, B., Goldreich, O. , Kushilevitz, E., Sudan, M. (1995) Private information retrieval, *Proceedings of IEEE 36th Annual Foundations of Computer Science*, Milwaukee, WI, USA, 1995, pp. 41-50.
10. Chor, B., Goldreich, O., Kushilevitz, E., Sudan, M. (1998) Private information retrieval. *Journal of the ACM* 45, 965-981.
11. Das, B. C., Amini, M. H., Wu, Y. (2025) Security and privacy challenges of large language models, *ACM Computing Surveys* 57:6 (June 2025)
12. Dignum, V., Dignum, F. (2025) Agentifying Agentic AI. *WMAC 2026 - AAAI 2026 Bridge Program on Advancing LLM-Based Multi-Agent Collaboration*. [arXiv:2511.17332](https://arxiv.org/abs/2511.17332)
13. Domingo-Ferrer, J., Bras-Amorós, M., Wu, Q., Manjón, J.A. (2009) User-private information retrieval based on a peer-to-peer community. *Data Knowl. Eng.* 68(11): 1237-1252.
14. Freij-Hollanti, R., Gnilke, O. W., Hollanti, C., Karpuk, D. A. (2017) Private information retrieval from coded databases with colluding servers, *SIAM J. Appl. Algebra Geom.* 1 647-664.
15. Goldwasser, S, Micali, S (1984) Probabilistic encryption, *Journal of Computer and System Sciences*, Volume 28, Issue 2, 270-299.
16. Gu, Z., Lanti, D., Corcoglioniti, F., Panfilo, M. D., Mosca, A., Calvanese, D., Xiao, G. (2025) Ontology-based data federation and query optimization, *Knowledge Based Systems* 329 114216.
17. Holzbaur, L., Freij-Hollanti, R., Li, J., Hollanti, C. (2022) Toward the capacity of private information retrieval from coded and colluding servers, *IEEE Trans. Inform. Theory* 68 517-537.
18. Holzbaur L., Hollanti, C., Wachter-Zeh, A. (2020) Computational Code-Based Single-Server Private Information Retrieval, *IEEE International Symposium on Information Theory (ISIT)*, Los Angeles, CA, USA, pp. 1065-1070.
19. Huang, C., Simitci, H., Xu, Y., Ogus, A., Calder, B., Gopalan, P., Li, J., Yekhanin, S. (2012) Erasure Coding in Windows Azure Storage, *USENIX ATC 2012* 15-26.

20. Isaev, P., Hammer, P. (2023) Memory System and Memory Types for Real-Time Reasoning Systems, Proc. AGI 2023, LNAI 13921 147-157.
21. Kotseruba, I., Tsotsos, J. K. (2020) 40 years of cognitive architectures: core cognitive abilities and practical applications, Artificial Intelligence Review 53 17-94.
22. Kushilevitz, E., Ostrovsky, R. (1997) Replication is not needed: single database, computationally-private information retrieval. In: Proc. of the 38th Annual IEEE Symposium on Foundations of Computer Science, pp. 364-373.
23. Laird, J. (2012) The Soar Cognitive Architecture. MIT Press. <https://doi.org/10.7551/mitpress/7688.001.0001>
24. Ostrovsky, R., Skeith, W.E. (2007) A Survey of Single-Database Private Information Retrieval: Techniques and Applications. In: Okamoto, T., Wang, X. (eds) Public Key Cryptography - PKC 2007. PKC 2007. Springer Lecture Notes in Computer Science, vol 4450.
25. Pindado, Z., Spendlhofer, T., Allam, M., Mehta, P., Martens, L., Peña, A. J. (2026) SoK of Private Deep Neural Network Inference with Approximate Fully Homomorphic Encryption, <https://eprint.iacr.org/2026/047>
26. Rando, J., Zhang, J., Carlini, N., Tramèr, F. (2025) Adversarial ML Problems Are Getting Harder to Solve and to Evaluate, Proc. DLSP 2025.
27. Samarati, P. (2001) Protecting Respondents' Identities in Microdata Release, IEEE Trans. on Knowledge and Data Engineering, 13:6 1010-1027.
28. Sathiamoorthy, M., Asteris, M., Papailiopoulos, D. S., Dimakis, A. G., Vadali, R., Chen, S., Borthakur, D. (2013) XORing Elephants: Novel Erasure Codes for Big Data, VLDB Endow. 6 325-336.
29. Shah, N.B., Rashmi, K.V., Ramchandran, K (2014) One extra bit of download ensures perfectly private information retrieval, 2014 IEEE International Symposium on Information Theory, Honolulu, HI, USA, 2014, pp. 856-860.
30. Tajeddine, R., Gnilke, O. W., El Rouayheb, S. (2018) Private information retrieval from MDS coded data in distributed storage systems, IEEE Transactions on Information Theory, vol. 64, no. 11, pp. 7081-7093.
31. Torra, V. (2022) Guide to data privacy: Models, Technologies, Solutions, Springer.
32. Torra, V., Bras-Amorós, M. (2026) Memory poisoning and secure multi-agent systems, arXiv:2603.20357
33. Torra, V., Navarro-Arribas, G., Sanchez-Charles, D., Muntés-Mulero, V. (2017) Provenance and privacy, Proc. MDAI 2017.
34. Verma, N., Hollanti, C. (2024) Code-Based Single-Server Private Information Retrieval: Circumventing the Sub-Query Attack, 2024 IEEE International Symposium on Information Theory (ISIT), Athens, Greece, 2024, pp. 2880-2885.
35. Wooldridge, M. (2006) An introduction to multiagent systems, John Wiley and Sons.
36. Wu, Y., Liang, S., Zhang, C., Wang, Y., Zhang, Y., Guo, H., Tang, R., Liu, Y. (2025) From human memory to AI memory: a survey on memory mechanisms in the era of LLMs, arXiv:2504.1595v2.
37. <https://www.mdai.cat/code/>
38. <https://www.swi-prolog.org/pldoc/man?section=prologjiti>
39. <https://sicstus.sics.se/performance.html>

AFTERNOON SESSION 1

Credentials, Anonymity and Secure Protocols

SoK: Data Minimization in Verifiable Credentials

Norbert Henseler^{1,2}[0009-0000-5738-7187], Robin Doss¹[0000-0001-6143-6850], Anh Dinh¹[0000-0002-8158-3636], Marthie Grobler²[0000-0001-6933-0145], and Tingmin Wu²[0000-0003-0626-3576]

¹ Deakin University, Australia <https://www.deakin.edu.au>

² CSIRO, Australia <https://www.csiro.au>

Abstract Digital identity technologies are increasingly used to replace physical or paper-based credentials in online and offline interactions, enabling individuals to prove facts about themselves in secure, tamper-resistant, and privacy-preserving ways. Within this field, Data Minimization (DM) is a core privacy-by-design principle aimed at ensuring that only strictly necessary information is shared. In practice, however, existing Verifiable Credential (VC) implementations include unnecessary attributes or excessive data, creating tension with regulatory expectations and privacy goals. Prior surveys acknowledge DM but do not examine it systematically across the full VC lifecycle. This study addresses this gap by categorizing and synthesizing existing DM techniques within a role- and lifecycle-aware framework. Our analysis shows that DM in VC systems is not a single design choice but a combination of Selective Disclosure (SD), Rule-Based DM, and Machine Learning-Based techniques, each with distinct trade-offs across roles, lifecycle stages, and application domains. This Systematization of Knowledge (SoK) provides a clear basis for selecting and combining DM techniques in practice.

Keywords: Verifiable Credentials · Data Minimization · Selective Disclosure · Credential Lifecycle · Role- and Lifecycle-Aware Framework

1 Introduction

Verifiable Credentials (VCs) are World Wide Web Consortium (W3C)-standardized digital identity documents such as driver’s licenses, passports, and academic degrees that users can store in wallet applications and present for verification [55]. A core privacy feature of the W3C VC standard is Data Minimization (DM), which means that only the minimum information necessary to fulfill a specific purpose should be disclosed.

By enabling Selective Disclosure (SD) and Verifiable Presentations (VPs), VCs support privacy-preserving interactions across education, healthcare, finance, and the Internet of Things (IoT) [4,13,18,19,30,45,55]. However, real-world VC implementations fall short because issuers include additional attributes, and verifiers request full credentials rather than only the minimum required attributes [1,31]. Prior work has examined SD, Self-Sovereign Identity (SSI), Machine Learning (ML), and regulatory aspects of DM in VCs, but does not provide

an integrated role- and lifecycle-aware systematization spanning issuer, holder, and verifier responsibilities. The literature proposes several DM techniques, including SD, Rule-Based DM, and ML-Based techniques, but these are usually studied in isolation rather than within a VC- and lifecycle-aware framework.

This Systematization of Knowledge (SoK) synthesizes SD-based, Rule-Based, and ML-Based DM. The initial search covered January 2020 to July 2025 and was updated in August 2026. We paid explicit attention to the three core VC roles: issuer, holder, and verifier. The analysis examines how DM can be achieved across the VC lifecycle and is guided by the following research questions:

- **RQ1:** Which DM techniques have been proposed, and how can they be systematically organized into a role- and lifecycle-aware framework that includes the perspectives of issuers, holders, and verifiers?
- **RQ2:** In which application domains have these techniques been applied, and how transferable are they across domains?

In response, this SoK makes three main contributions:

- **C1:** A taxonomy of DM techniques for VCs (Section 5.1).
- **C2:** A role- and lifecycle-aware conceptual framework that organizes DM techniques from the perspectives of issuers, holders, and verifiers, intended to support analysis, comparison, and future system design (Section 5.2).
- **C3:** A mapping of DM techniques across key domains (Section 4.2).

2 Background and Conceptual Foundations

This section introduces VCs, outlining their core definitions, roles, ecosystem, and lifecycle. It examines privacy provisions in VC standards, introduces our VC threat model, and conceptualizes DM across the VC lifecycle.

Running example. In the proof-of-age use case, we illustrate the responsibilities of the relevant roles. A state government agency issues a digital driver's license to the holder's wallet. This VC is signed by the issuer and includes a license identifier, date of birth, validity period, license class, and restrictions. At checkout, the beer and wine store (verifier) requests proof-of-age (aged 18 or over) from the customer (holder). The holder creates a Verifiable Presentation (VP) that discloses only the claim " $age \geq 18$ ". The store verifies the government agency's signature and the driver's license status, then approves the sale.

2.1 Verifiable Credentials: Definition, Roles, and Ecosystem

A VC is a tamper-evident digital credential that contains claims about a subject and can be cryptographically verified. In the W3C data model, issuers create and sign VCs, holders store them and generate VPs, and verifiers validate presented claims [55]. Issuers may revoke or reissue VCs, and verifiers may log verification events.

VC deployments may use supporting components such as a Verifiable Data Registry (VDR) and Decentralized Identifiers (DIDs). DIDs are optional and are not required by the W3C VC data model [54,55]. Together with the core roles, such supporting components form a VC ecosystem [38].

2.2 Privacy in the VC Standard

The VC standard supports privacy-oriented designs in which verifiers can request information at the claim level rather than full credentials and holders can selectively disclose attributes from their VCs. For example, the VC data model recommends that issuers include only essential claims and that holders present only the claims a verifier needs. The standard defines a "spectrum of privacy", ranging from "highly correlatable" to "non-correlatable". In our running example, the verifier can request an age predicate without revealing the holder's date of birth. However, this may still enable holder identification [55]. Cryptographic proofs can further reduce linkability, depending on the design of the credential [55]. DIDs also affect privacy in VC ecosystems. When holders use different DIDs across transactions, presentations made to one verifier become harder to link to those made to other verifiers. This does not guarantee unlinkability, but it can reduce the risk of correlation.

2.3 DM and Role-Based VC Threat Model

In this paper, DM is treated as a privacy principle that limits the collection, storage, and disclosure of Personally Identifiable Information (PII) to what is necessary for a specific purpose. DM techniques are the approaches used to operationalize the DM principle throughout the VC lifecycle, including credential issuance, wallet storage, and presentation. DM mechanisms are used to implement the techniques where relevant.

The threat model distinguishes between internal and external threat sources. Internal threats arise from actors within the VC ecosystem, primarily issuers and verifiers that deviate from intended DM practices, while external threats originate from actors who compromise, intercept, or replay private data. Replay is DM-relevant because intercepted presentations may be reused without the holder's consent, resulting in unintended disclosure. The holder is treated as the victim rather than a malicious actor, and the model focuses on issuer- and verifier-side behaviors that violate DM, with the VDR considered a supporting system rather than a primary threat source. Based on this setting, we define six DM-relevant threats (T1-T6) across VC issuance, wallet storage, and presentation. Table 1 captures issuer over-collection (T1), wallet breaches by an external adversary (T2), verifier-side correlation (T3), excessive data requests by the verifier (T4), replay of intercepted presentations (T5), and collusion of multiple verifiers or issuers and verifiers (T6).

Table 1. Adversarial Settings for the VC Ecosystem

Threat	Primary Threat Source	Adversarial Setting	Threat Description
T1	Issuer	Issuer Over-Collection	The issuer adds unnecessary attributes to a VC, increasing the disclosure risk later in the VC life-cycle.
T2	External adversary	Wallet Breach	VC data is exposed from the holder's local or cloud wallet.
T3	Verifier	Presentation Correlation	The verifier links multiple VPs to re-identify the holder across presentations.
T4	Verifier	Overreaching Request	The verifier requests unnecessary attributes.
T5	External adversary	Presentation Replay	Intercepted VPs are reused or replayed beyond the holder's intended usage.
T6	Colluding issuers and verifiers	Cross-Transaction Collusion	Multiple verifiers or colluding issuers and verifiers share presentation data to re-identify holders or infer additional information.

2.4 VC Lifecycle and Conceptualizing Data Minimization

Applying the big data lifecycle [43] to VCs begins with the issuer deciding, based on the anticipated usage scenarios [55], which attributes to include in the VC. The issuer then creates and signs the VC from source records. The holder stores the VC in a wallet, where device-level security and local policies protect its integrity and confidentiality. A VC is immutable once issued. Operationally, updates are implemented by reissuing the VC. The verifier requests information for a specific purpose, and the holder applies SD to disclose only the required attributes. In the running example, the holder may prove they are "18 or over" at a beer and wine store without revealing their date of birth. The verifier then validates the presentation prior to using the data. Finally, VCs may expire or be revoked by the issuer.

Conceptualizing DM across the VC lifecycle involves examining how much identifying information is shared or collected at different stages of the lifecycle, and how effectively the system mitigates vulnerabilities while preserving its intended purpose. We assess DM across issuance, wallet storage, and presentation, because these are the stages at which claims are created, stored, requested, disclosed, or reused. Yang et al. distinguish adequacy, relevance, and necessity. Requested data must contribute to the purpose, be related to it, and be limited to the least revealing information required to achieve it [58]. In the running example, the age predicate may satisfy the purpose, whereas requesting the date of birth requires additional justification.

A VC with fewer attributes at issuance reduces downstream exposure, because unnecessary claims are not in the VC at all. Reducing the number of disclosed claims may lower exposure, but privacy risk can remain high if the disclosed claims are highly identifying. The VC threat model in Section 2.3 helps locate where each risk appears. If an issuer avoids unnecessary claims, the leakage risk in later stages shrinks because the data are not present in the VC. Verifier correlation affects presentation risks and replay, while collusion affects reuse and aggregation risks across transactions.

3 Methodology

We used Kitchenham and Brereton’s guidance to structure the review design [35], and the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) framework to document the search and selection process [40].

3.1 Sources and Search Strategy

We identified IEEE Xplore and Scopus as primary databases. IEEE was selected for its strong coverage of security research, and Scopus for its multidisciplinary coverage. We also conducted forward and backward citation searches. Our search string was designed as an intentionally broad query to capture both VC-specific studies and transferable DM techniques. We searched titles, abstracts, and keywords. Our search combined three concept groups: (A) VCs, decentralized identity, DM; (B) privacy, regulatory terminology; and (C) selective disclosure, zero-knowledge proofs (ZKPs), ML. We combined the three concept groups (A, B, C) into two logical blocks and used the structure A AND (B OR C). The conceptual structured query is shown in Table 2. Relevance to DM in VC ecosystems was then enforced through title and abstract screening and full-text assessment.

3.2 Inclusion, Exclusion, and Screening

We included peer-reviewed conference or journal papers published between January 2020 and August 2026. The initial search was conducted in July 2025 and updated in August 2026. Our search retained papers that proposed, analyzed, or evaluated DM techniques in VC ecosystems, as well as studies that applied DM techniques transferable to VC needs. We excluded papers that discussed SSI architectures, DIDs, blockchain, or privacy in general, unless they contributed to DM in VC ecosystems. Studies retrieved using general privacy, machine learning, zero-knowledge, or data-minimization terminology were retained only when full-text screening showed a transferable contribution to DM in VC ecosystems.

The initial database search returned 1,169 papers. We used Covidence³ for deduplication, screening, and study selection. After deduplication, 892 papers remained. The titles and abstracts of these were screened against the inclusion criteria, and the remaining papers were assessed at full text. Of the 352 full-text papers assessed, 298 papers were excluded because they met one or more exclusion criteria. The remaining 54 papers met all criteria and were included in the qualitative synthesis. The supplementary search in August 2026 identified three additional studies. The combined synthesis therefore contains 57 studies.

3.3 Data Extraction and Synthesis

For each of the 57 papers, we systematically extracted and analyzed information on groups of DM techniques, the cryptographic mechanisms, evaluation evidence,

³ <https://www.covidence.org/>

Table 2. Conceptual Boolean Search String.

Concept Group	Search Terms
A	"verifiable credential*" OR "attribute-based credential*" OR "anonymous credential*" OR "digital credential*" OR "decentralized identity" OR "data minimization" OR "data minimisation"
B	"privacy" OR "privacy-preserving" OR "anonymity" OR "unlinkability" OR "data protection" OR "GDPR" OR "CCPA" OR "CPRA"
C	"selective disclosure" OR "zero-knowledge" OR "data minimization" OR "data minimisation" OR "machine learning" OR "AI" OR "reinforcement learning"
Full query: A AND (B OR C)	

and regulatory basis. We recorded how studies translated legal principles and the application domains or use cases in which they were studied. Recurring approaches, such as SD with ZKPs, were examined across multiple studies to assess how their effectiveness is evaluated. To contextualize these findings, each study was mapped to an application domain and one or more use cases.

4 Systematization of Techniques

We extracted key information related to the research questions and systematically analyzed the DM techniques and use cases to identify common categories. Applicable studies were counted in multiple categories. Across the selected studies, we identified three groups of DM techniques: VC-specific SD, transferable Rule-Based DM, and ML-Based DM.

4.1 DM Techniques and Usage in the VC Lifecycle (RQ1)

Selective Disclosure. VCs use SD as their primary DM technique. SD allows the holder to reveal only the attributes required by the verifier [18,19,45,55]. In our review, 28 studies (49% of 57) use some form of SD, making it the most common approach. The reviewed papers group SD in VCs into three mechanisms: atomic credentials, commitment-based schemes, and ZKP-based schemes.

Atomic credentials allow the issuer to create a separately signed credential for each claim [18]. They appear in five selected studies. Three earlier studies describe the approach [18,19,45], while Buldini et al. and De Salve et al. implement atomic credentials as experimental baselines [5,14]. Their evaluations show fast VP generation but increased storage, verification, and revocation costs as the number of claims grows. Additionally, combining multiple atomic credentials within a single VP may result in invalid presentations [45]. We therefore treat atomic credentials as an experimental baseline rather than a scalable strategy.

Commitment-based schemes, covered in seven studies, hide undisclosed attributes until the holder chooses to reveal them. Typically, the issuer commits to each attribute in the VC, and the holder opens only the commitments required by the verifier. These schemes are conceptually simpler and can reduce cryptographic complexity, but require additional handling of committed data and presentation logic [19,45].

ZKP-based SD is the most common mechanism, covered by 19 studies. Schemes based on Camenisch-Lysyanskaya (CL) or Boneh-Boyen-Shacham+ (BBS+) signatures allow holders to prove selected claims without revealing all VC data [18,45]. Recent work improves efficiency: Guo et al. [27] use partial blind signatures with a Zero-Knowledge Succinct Non-Interactive Argument of Knowledge to achieve constant-size proofs, while Ramić et al. [45] combine Boneh-Lynn-Shacham signatures, Merkle trees, and Bulletproofs for multi-credential presentations and range proofs. Architecturally, these mechanisms are used in systems for decentralized identity, IoT data sharing, and VCs, enabling vehicles to share attributes while maintaining privacy [21,59,60]. Yang et al. complement these studies by linking VC formats to the GDPR requirements of adequacy, relevance, and necessity. Their analysis finds that most common formats support claim-level SD while BBS+-based credentials and Anonymous Credentials additionally support predicate proofs.

Rule-Based. Rule-Based DM relies on predefined, deterministic policies or constraints to limit the personal data collected, processed, or disclosed [9]. We distinguish three minimization mechanisms in the reviewed literature: horizontal DM (hDM), vertical DM (vDM), and transformative DM (tDM). We treat these as mechanisms and Rule-Based DM as a technique for enforcing them.

hDM focuses on reducing the number of data records processed or disclosed [33,32,50]. By lowering disclosure volume, hDM reduces exposure and the potential for post-breach identification, although highly detailed records may still allow holder identification. In IoT settings, related approaches limit the volume or frequency of data exposed or processed [33,50]. Jaiswal et al. demonstrate how this principle can be applied in combination with Federated Learning [32].

vDM focuses on collecting less detailed data, limiting or generalizing unnecessary details. In a VC implementation, this means that issuers and holders should protect individual privacy at the transaction level by limiting the number of claims disclosed, with verifiers excluding unnecessary attributes rather than requesting full credential data. Returning to our running example, *hDM* limits the claims presented to one identity document, while *vDM* restricts the requested information to an age predicate rather than a full date of birth.

tDM goes a step further by transforming data to reduce its sensitivity while preserving usefulness. Common approaches include generalization, such as recording age ranges instead of exact ages or approximate locations rather than precise coordinates. Pu et al. implement facial DM using ML-assisted tDM [42]. Park [41] presents the "Protocol for Data Minimization" (DtMin) in healthcare, which shares only patient-consented and requester-required medical attributes in readable form while keeping the remaining data encrypted. Unlike SD, which enables selective presentation at verification time, DtMin controls data exposure through transformation and policy enforcement. Garofalo et al. use *tDM* to share health data with minimized re-identification risks [25]. tDM strengthens privacy by making unnecessary data inaccessible, but it introduces additional complexity and requires careful trade-offs between privacy and functionality.

Machine Learning. We found no VC-specific study where an ML system dynamically formulates minimal verifier requests within the standardized VC workflow. We therefore treat ML-Based DM as a transferable rather than an established VC-specific technique. Unlike deterministic Rule-Based approaches, ML-Based methods may help estimate the necessity of attributes based on context or task outcomes rather than on fixed policies.

Audit methodologies assess whether previously collected attributes affect outcomes, supporting assessment in retrospect of minimization and evaluation of initial data-collection strategies [23]. Rastegarpanah et al. [46], for example, test whether inputs are necessary by replacing them with constants and observing whether outcomes change, arguing that unchanged predictions indicate unnecessary inputs. Similarly, scaling-law-based approaches such as the Framework for Inhibiting Data Over-Collection use scaling laws to identify diminishing returns from additional data collection [49]. While these mechanisms help to prevent over-collection, they do not generate minimal VC requests.

Real-time filters help users identify, withhold, or generalize sensitive data before disclosure. SecureCheck introduces a privacy-enhancing framework to reduce data oversharing during verification by using user-verifier contracts to control shared information [31]. Tran et al. [52] show that accurate inference may require as little as 10% of the attributes, and Zhou et al. [62] use a large language model to flag PII before sharing. In a VC implementation, such approaches could support holder disclosure decisions, but current work does not integrate ML into VC presentation workflows.

Federated learning supports DM by ensuring that only model parameters are shared, keeping sensitive data local [17,26,36].

Taken together, the VC literature is holder-centric and focuses on SD, while support for issuers and verifiers remains limited. Rule-Based DM techniques can help align issuers and verifiers with DM requirements. ML-related work suggests potential future support for VC workflows, but there is no VC-specific evidence that these methods can help issuers minimize VC schemas or help verifiers formulate minimal data requests.

4.2 Application Domains and Transferability (RQ2)

We grouped the application domains as proposed by W3C [53] and added a column for domain-independent studies. Table 3 contains representative studies and summarizes how the selected studies apply DM techniques across application domains. In education, VCs enable selective disclosure of data (attributes and predicates) from diplomas or transcripts, allowing graduates to prove qualifications without revealing additional personal data and supporting revocation where necessary [45]. Hartmann et al. [28] present another example where collected student click data revealed sensitive personal information, including academic achievement and demographic data, highlighting the challenges of DM in practice. Financial institutions use ZKP-based range proofs to meet requirements such as know-your-customer (KYC) [24]. Retail use cases align closely

with our running example, in which the verifier requires eligibility rather than the holder’s identity [18,45,47].

In healthcare, DM is important due to the high sensitivity of the data [41]. The DtMin Protocol, for example, supports medical information sharing between healthcare providers and health record providers, ensuring disclosure of patient-consented and required attributes. In neuropsychological studies, a mobile health (mHealth) app prioritizes data aggregation on the device, avoiding raw sensor data collection [22]. Specialized auditing frameworks, such as MINDAEXT (for browser extensions), are adaptable to multiple privacy regulations. Although transferring these frameworks to domains such as IoT may require adjustments to data collection techniques, their underlying approach remains adaptable [37]. Furthermore, techniques such as vDM remain limited to structured tabular data (e.g., census data, loan data) [3,50]. Another challenge to broader adoption is the lack of flexible, efficient implementations for diverse domain requirements. Recent research has begun to close this gap by introducing architectures suited to diverse use cases [24].

Table 3. Coverage of DM Techniques and Mechanisms across Application Domains.

Technique	Mechanism	Application Domain							Domain Independent
		Education	Retail	Finance	Healthcare	Professional Credentials	Legal Identity	IoT	
SD	Atomic credentials	-	-	-	-	-	-	-	[18,19,45,5,14]
	Commitments/hash-based	[34]	[18]	-	-	-	[13]	-	[19,5,14,58]
	ZKPs	[45]	[18]	[24]	-	[45]	[4,27,29,30] [39,47,57,59]	[7,12,20,61]	[19,5,14,58]
Rule-Based	hDM	-	-	-	[32]	-	-	[33]	[50]
	vDM	-	-	-	[31]	-	[1,11]	[9,10]	[50]
	tDM	-	-	-	[22,25,41]	-	-	-	[42]
ML	Audit	[28]	-	-	-	-	-	-	[37,23,46,49]
	PII Filtering	-	-	-	-	-	-	-	[52,62]
	Federated learning	-	-	-	[17,26,36]	-	-	-	[32]

"[]": Indicates coverage; "-": indicates no coverage among the selected studies.

Studies may appear under multiple mechanisms. Consequently, mechanism counts do not add up to the number of unique studies.

Overall, the transferability of these techniques varies with the level of abstraction. Core cryptographic mechanisms, such as ZKPs, are highly transferable across VC domains because DM requires holder control over shared claims. Rule-Based mechanisms transfer when domain requirements can be generalized and reused in different domains. ML-Based mechanisms remain the least mature for VCs because there is no specific research on their use in VC workflows.

5 Discussion

5.1 Evaluation of DM Techniques Across Domains (C1, C3)

Table 4 shows our taxonomy of DM techniques and compares the identified DM mechanisms using six criteria. *Privacy impact* indicates how much a technique reduces attribute disclosure and linkability. *Efficiency* refers to the additional

computational and communication costs incurred at issuance, presentation, or verification. High efficiency means low computational and communication overhead. *Scalability* indicates whether the technique remains usable as the number of VCs, presentations, or actors grows. *Domain transferability* refers to evidence that the technique can be used across multiple domains. *Regulatory alignment* refers to whether the technique can support DM responsibilities under the Australian Privacy Act [2], the General Data Protection Regulation [16], and the California Consumer Privacy Act [6]. Following Yang et al. [58], high alignment requires limitation of data quantity to what is sufficient for the purpose. Medium indicates that the technique can reduce data, but depends on external policies. Low alignment indicates indirect or retrospective support. *Quantum resilience* shows whether the underlying security mechanisms are likely to withstand threats created by future quantum computers. The high, medium, and low ratings summarize how the reviewed work highlights trade-offs among different DM techniques and mechanisms. These ratings synthesize the evidence discussed in Section 4.1 and are indicative rather than precise. They should not be taken as a benchmark for DM techniques.

Atomic credentials are rated low for scalability because the number of credentials, signatures, and revocation operations grows with the number of claims [5,14]. De Salve et al. show that hash-tree presentation size and processing costs can increase considerably with the number of disclosed claims [14]; hence, scalability for hash-based commitments is rated medium. ZKP-based mechanisms justify high privacy impact because they enable proofs about attributes or predicates without revealing the underlying data [7,19,56,61]. However, they come with increased complexity for proving and verification and therefore score lower on efficiency, with scalability remaining medium due to design and implementation trade-offs [19]. Regulatory alignment is high because minimized disclosure supports necessity and purpose limitation. Column n reports the number of selected studies supporting each mechanism. The 19 ZKP-related SD studies are regrouped into nine SD-signature studies and 10 broader ZKP studies, while commitments remain a separate category. The comparison shows that SD techniques and ZKP-based mechanisms provide the strongest VC support for minimization at presentation time, but often involve higher cryptographic complexity. Rule-Based DM better supports issuer- and verifier-side minimization. ML-Based DM techniques remain the least mature for VC workflows. We found no VC-specific study in which ML-Based techniques help to formulate minimal verifier requests or issuer schemas. Table 4 reinforces the view that DM in VC ecosystems is a lifecycle property with responsibilities for the issuer, holder, and verifier.

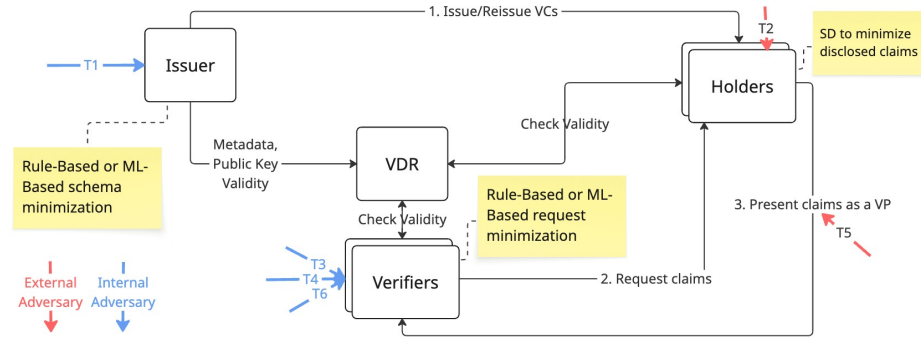
5.2 Role- and Lifecycle-Aware DM Framework (C2)

A key contribution of our work is a lifecycle-aware framework that maps DM techniques to the VC lifecycle phases and to the roles of key actors, as shown in Figure 1. We organize the framework by the three main lifecycle phases (issuance, storage, presentation), highlighting how each participant (issuer, holder, verifier) can support DM at each stage. This mapping clarifies that privacy-by-design in

Table 4. Comparative Evaluation of the Taxonomy.

	Technique	Mechanism	n	Criteria				
				Privacy Impact	Efficiency	Scalability	Domain Transferability	Regulatory Alignment
Taxonomy	SD	Atomic credentials	5	Medium	Medium	Low	Low	Medium
		Commitments/hash-based	7	Medium	High	Medium	Low	Medium
		SD Signatures	9	High	Medium	Medium	Low	Medium
		ZKP	10	High	Low	Medium	High	Medium
	Rule-Based	hDM	3	Low	High	High	Medium	Medium
		vDM	6	High	High	Medium	Medium	High
		tDM	4	High	Medium	Medium	Medium	High
	ML-Based	Audit	5	Low	High	High	Low	Low
		PII Filtering	2	Medium	Medium	Low	Low	Medium
		Federated learning	4	Low	Low	Medium	Low	Low

VC ecosystems depends on decisions made by different actors at different lifecycle stages rather than on holder-side SD alone.

**Figure 1.** Role- and Lifecycle-Aware Framework of DM Techniques for VCs

Issuance Phase, Issuer Responsibilities. In the issuance phase, the issuer creates and signs the VC. Here, DM means issuing VCs that include only the attributes necessary for the intended purpose [55], avoiding unnecessary detail (mitigating threat T1). While an issuer’s signature establishes trust, the choice of signature scheme is critical for unlinkability, performance, and security [19]. If the issuer signs the entire VC as a single object, later verification may require disclosure of the full credential, thereby undermining SD. By contrast, applying standard digital signature primitives at the claim level allows claims to be proven independently and supports SD [17,41,45,55]. Overall, the issuer’s responsibility is to embed privacy-by-design into the VC’s design. Issuers should design VCs with minimal, flexible schemas and signature and proof constructions that allow the holder to create minimal, privacy-preserving presentations.

Storage Phase, Holder Responsibilities. After issuance, the VC is stored on the holder’s device, typically in a digital wallet. The wallet protects data at rest, aligning with Cavoukian’s principle of end-to-end security [8]. While holders

must retain VCs for future use, a privacy-by-design wallet would mitigate the risk of compromise (Threat T2). For example, a wallet might store only frequently used VCs and archive or delete expired or unused VCs, thereby reducing the amount of PII available to an attacker if the device is compromised. Our survey did not identify any implementations of storage-phase DM. We identify this as a direction for future research.

Presentation Phase, Holder and Verifier Responsibilities. At the start of the presentation phase, a verifier requests information, and a holder proves facts from their VCs. For verifiers, DM entails requesting only the data necessary to provide the service. However, our analysis found that verifiers lack tools to systematically minimize requests, with over-requesting remaining common practice [31,52]. An ideal verifier would automatically derive the minimal data requirements from service needs; for example, requesting proof that the account balance exceeds a required threshold rather than a full financial report. While ML-Based approaches to DM have been discussed in the literature [46,52], we found no VC-specific proof-of-concept that uses ML to formulate or minimize verifier requests within a standardized VC workflow, indicating a gap. Verifiers are also responsible for minimizing data retention after presentation, ideally by deleting or anonymizing the data received. Yet, the reviewed studies provide no technical enforcement for such practices, relying instead on policy and organizational discipline.

A holder responds to a verifier’s request by presenting the requested attributes with a cryptographic proof, allowing the verifier to make an authorization decision based on the policy. While such proofs enable privacy-preserving verification of required claims, they do not themselves enforce access control. The holder’s role in DM is therefore to use SD, typically via VPs, to disclose the minimum information at presentation time [31,59]. The presentation phase is where the privacy-utility trade-off is resolved: holders seek to minimize disclosure, while verifiers must avoid excessive demands. Our DM framework shows that without verifier cooperation, holder-side DM remains limited, highlighting a gap in verifier-side DM techniques that support holder privacy.

DM Framework. Overall, DM is an ecosystem property. Issuers constrain schemas, holders minimize disclosures in VPs, and verifiers constrain requests and retention. Current work remains weakest on issuer schemas, verifier requests, and quantum-resilient designs.

6 Related Work

Prior reviews treat DM as a privacy and security problem rather than as a life-cycle problem. Mazzocca et al. [38] survey DIDs and VCs, list security threats, implementations, and real-world applications, but they do not analyze DM techniques or their trade-offs. Schardong et al. [48] conduct a systematic review of SSI studies and propose an SSI taxonomy, but they do not focus on DM. Ramić et

al. [44] review SD techniques in Anonymous Credentials, attribute-based credentials, and VCs. Their review is helpful for designing cryptographic mechanisms, but it focuses on SD rather than DM throughout the VC lifecycle. Staab et al. [51] systematize DM in ML and classify techniques such as federated learning, differential privacy, synthetic data, data augmentation, and feature selection as part of raw data pipelines. In contrast, VC ecosystems are based on the roles of issuers, holders, and verifiers, which interact through cryptographically signed, standardized VCs, defined in the W3C data model [55]. Recent studies provide detailed experimental evidence for SD [5,14] and Yang et al. connect GDPR requirements to credential formats [58]. Our SoK integrates these perspectives into a role- and lifecycle-aware systematization of DM techniques, evidence, and research gaps.

7 Gaps and Future Research

DM support for issuers is essential to achieving privacy-by-design, which assumes privacy by default even when issuers cannot anticipate all of the verifier use cases. The open challenge is to design VC schemas that avoid over-collection at issuance while remaining adaptable to changes.

DM support for holders continues to rely primarily on SD. A key gap concerns the transition to quantum-secure signature schemes that remain computationally efficient and compact. Current schemes involve trade-offs between functionality and quantum resilience [19,44,41].

DM support for verifiers remains particularly limited. Verifiers generally lack mechanisms to formulate minimal, purpose-bound requests, resulting in systematic over-requesting. Even in our running example, the reviewed literature provides no VC-specific technical support for generating minimal verifier requests. Promising directions are formal policy languages or decision-support tools that translate service requirements into minimal claim requests.

Performance and scalability constraints further influence practical deployment. Buldini et al. and De Salve et al. quantify performance, storage, and communication trade-offs among atomic, hash-based, SD signatures, and ZKP-based DM mechanisms. However, they evaluate single mechanisms rather than VC workflows. Future studies should evaluate end-to-end VC workflows including performance across issuance, wallet storage, and presentation under realistic workloads.

Usability and adoption present ongoing challenges. VC wallets are often difficult to use, leading to low adoption rates, which are intensified by limited cross-platform integration [38]. Improved interaction design, user guidance, and stronger integration standards are needed.

Revocation and post-disclosure control raise unresolved privacy concerns. The goal is to enable verifiers to check the revocation status without compromising holder unlinkability [48]. After disclosure, holders generally lack technical control over the subsequent use of shared data [15,48].

8 Limitations

This study is a time-bound snapshot of DM in VCs and should not be generalized beyond VC ecosystems without further validation. The review prioritizes peer-reviewed literature, W3C standards, and reproducible database searches, which may omit relevant preprints, industry reports, or implementation repositories. The analysis relies on evidence reported by the selected studies rather than a benchmark conducted by this SoK. Although Buldini et al. and De Salve et al. provide reproducible comparisons of DM mechanisms [5,14], their experimental configurations differ and do not provide end-to-end VC lifecycle performance. Yang et al. provide a detailed GDPR analysis [58]; our comparison of the Australian Privacy Act and the CCPA focuses on technical support rather than legal compliance.

9 Conclusion

This SoK critically examined DM techniques across the VC lifecycle, clarifying the objectives, assumptions, and trade-offs of existing approaches. Our main finding is that DM support in VC ecosystems remains uneven. Research has concentrated on holder-side SD at presentation, while issuer-side minimization at credential design and creation, verifier-side request minimization, and post-quantum cryptographic readiness are comparatively underdeveloped. This SoK lays a foundation for more privacy-preserving VC systems by situating these techniques within the W3C VC standard and legally grounded DM principles. By identifying where current approaches are mature enough and where they fall short, the review offers a realistic account of how DM can be operationalized across the VC lifecycle. We expect this to assist practitioners in distinguishing techniques that are mature enough for deployment from those that require further research or governance attention to meet regulatory requirements.

Acknowledgments. This research was supported by the CSIRO Industry PhD Program. The authors acknowledge the valuable contribution of Bupa.

References

1. Anciaux, N., Frittella, S., Joffroy, B., Nguyen, B., Scerri, G.: A new PET for Data Collection via Forms with Data Minimization, Full Accuracy and Informed Consent. In: EDBT. vol. 27, pp. 81–93 (2024)
2. Australia: Privacy act 1988 (1988), <https://www.oaic.gov.au/privacy/privacy-legislation/the-privacy-act>
3. Bargh, M.S., Meijer, R., Van Den Braak, S., Latenko, A., Vink, M., Choenni, S.: Embedding Personal Data Minimization Technologies in Organizations: Needs, Vision and Artifacts. In: ACM Int. Conf. Proc. Ser. pp. 71–79 (2021)
4. Braun, C.J., Käfer, T.: Attribute-based Access Control on Solid Pods using Privacy-friendly Credentials. In: CEUR Workshop Proc. vol. 3235 (2022)

5. Buldini, A., Mazzocca, C., Montanari, R., Uluagac, S.: Benchmarking selective disclosure mechanisms for verifiable credentials: A systematic comparison for security and privacy. *IEEE Trans. Inf. Forensics Secur.* **20**, 13205–13220 (2025)
6. California Legislature: California consumer privacy act of 2018 (2018), <https://oag.ca.gov/privacy/ccpa>
7. Casanova-Marqués, R., Torres-Sospedra, J., Hajny, J., Gould, M.: Maximizing privacy and security of collaborative indoor positioning using zero-knowledge proofs. *Internet of Things* **22** (2023)
8. Cavoukian, A.: Privacy by Design: The 7 Foundational Principles. Tech. rep. (2009)
9. Chen, Y., Alhanahnah, M., Sabelfeld, A., Chatterjee, R., Fernandes, E.: Practical Data Access Minimization in Trigger-Action Platforms. In: *USENIX Security*. pp. 2929–2945 (2022)
10. Chi, H., Zeng, Q., Du, X., Luo, L.: PFIREWALL: Semantics-Aware Customizable Data Flow Control for Smart Home Privacy Protection. In: *NDSS* (2021)
11. Da Silva Carvalho, N., Jabbarpour, J., Temple, L., Belacort, I.M., Barturen, U.I., Kortlander, M., et al.: A more inclusive Europe through personal data sovereignty in cross-border digital public services. In: *ACM Int. Conf. Proc. Ser.* pp. 63–71 (2023)
12. Daoudagh, S., Marchetti, E., Savarino, V., Bernabe, J.B., García-Rodríguez, J., Moreno, R.T., et al.: Data protection by design in the context of smart cities: A consent and access control proposal. *Sensors* **21**(21) (2021)
13. De Salve, A., Lisi, A., Mori, P., Ricci, L.: Selective Disclosure in Self-Sovereign Identity based on Hashed Values. In: *IEEE ISCC* (2022)
14. De Salve, A., Lisi, A., Cascino, M., Mori, P., Ricci, L.: Selective disclosure approaches in self-sovereign identity: An experimental comparison. *IEEE Access* **14**, 1152–1181 (2026)
15. Ernstberger, J., Lauinger, J., Elsheimy, F., Zhou, L., Steinhorst, S., Canetti, R., et al.: SoK: Data Sovereignty. In: *IEEE EuroS&P* (2023)
16. European Union: General data protection regulation (gdpr) (2018), <https://gdpr-info.eu>
17. Fdhila, W., Stifter, N., Judmayer, A.: Challenges and Opportunities of Blockchain for Auditable Processes in the Healthcare Sector. In: *LNBIP*. vol. 459, pp. 68–83 (2022)
18. Flamini, A., Ranise, S., Sciarretta, G., Scuro, M., Sharif, A., Tomasi, A.: A First Appraisal of Cryptographic Mechanisms for the Selective Disclosure of Verifiable Credentials. In: *SECRYPT*. vol. 1, pp. 123–134 (2023)
19. Flamini, A., Sciarretta, G., Scuro, M., Sharif, A., Tomasi, A., Ranise, S.: On cryptographic mechanisms for the selective disclosure of verifiable credentials. *J. Inf. Secur. Appl.* **83** (2024)
20. Fotiou, N., Pittaras, I., Chadoulos, S., Siris, V.A., Polyzos, G.C., Ipiotis, N., et al.: Authentication, Authorization, and Selective Disclosure for IoT Data Sharing Using Verifiable Credentials and Zero-Knowledge Proofs. In: *LNCS*. vol. 13782, pp. 88–101 (2023)
21. Fotiou, N., Kalos, V., Thomas, Y., Xylomenos, G., Siris, V.A., Polyzos, G.C.: Selective Content Disclosure using Zero-Knowledge Proofs. In: *GIIS*. pp. 52–56 (2022)
22. Gabel, A., Ertas, F., Pleger, M., Schiering, I., Müller, S.V.: Privacy by Design for Neuropsychological Studies Based on an mHealth App. In: *CCIS*. vol. 1400, pp. 442–467 (2021)
23. Ganesh, P., Tran, C., Shokri, R., Fioretto, F.: The data minimization principle in machine learning. In: *ACM conference*. pp. 3075–3093 (2025)

24. García-Rodríguez, J., Krenn, S., Bernal-Bernabe, J., Skarmeta, A.F.: Beyond selective disclosure: Extending distributed p-ABC implementations by commit-and-prove techniques. *Comput. Netw.* **248** (2024)
25. Garofalo, G., Van Hamme, T., Preuveneers, D., Joosen, W.: A Siamese Adversarial Anonymizer for Data Minimization in Biometric Applications. In: *EuroS&P Workshops*. pp. 334–343 (2020)
26. Geng, J., Kanwal, N., Jaatun, M.G., Rong, C.: DID-efed: Facilitating federated learning as a service with decentralized identities. In: *ACM Int. Conf. Proc. Ser.* pp. 329–335 (2021)
27. Guo, X., Zhao, Y., Cheng, Y., Jia, W., Jiang, Y.: AsCred: An anonymous credential system based on batch partial blind signature and polymath. *J. Inf. Secur. Appl.* **93** (2025)
28. Hartmann, J., Heuer, H., Breiter, A.: Data Leakage Through Click Data in Virtual Learning Environments. In: *LNI*. vol. P-322, pp. 135–146 (2022)
29. He, J., Ma, X., Zhang, D., Peng, F.: Supervised and revocable decentralized identity privacy protection scheme. *Security and Safety* **3** (2024)
30. Hoang, A.T., Ileri, C.U., Sanders, W., Schulte, S.: ZkSSI: A Zero-Knowledge-Based Self-Sovereign Identity Framework. In: *IEEE Blockchain*. pp. 276–285 (2024)
31. Hopkins, J., Rubio-Medrano, C.E.: SecureCheck: User-Centric and Geolocation-Aware Access Mediation Contracts for Sharing Private Data. In: *SACMAT*. pp. 53–58 (2024)
32. Jaiswal, A., Yigzaw, K.Y., Ozturk, P.: F-CBR: An Architecture for Federated Case-Based Reasoning. *IEEE Access* **10**, 75458–75471 (2022)
33. Jin, H., Liu, G., Hwang, D., Kumar, S., Agarwal, Y., Hong, J.I.: Peekaboo: A Hub-Based Approach to Enable Transparency in Data Processing within Smart Homes. In: *IEEE S&P* (2022)
34. Kaneriya, J., Patel, H.: A Secure and Privacy-Preserving Student Credential Verification System Using Blockchain Technology. *Int. J. Inf. Educ. Technol.* **13**(8), 1251–1260 (2023)
35. Kitchenham, B., Brereton, P.: A systematic review of systematic review process research in software engineering. *Inf. Softw. Technol.* **55**(12), 2049–2075 (2013)
36. Lieftink, N., Dos Ribeiro, C.S., Kroon, M., Haringhuizen, G.B., Wong, A., van de Burgwal, L.H.: The potential of federated learning for public health purposes: a qualitative analysis of GDPR compliance, Europe, 2021. *Eurosurveillance* **29**(38) (2024)
37. Ling, Y., Hao, Y., Wang, Y., Wang, K., Bai, G., Dong, J.S.: Essential or Excessive? MINDAEXT: Measuring Data Minimization Practices among Browser Extensions. In: *SANER*. pp. 964–975 (2024)
38. Mazzocca, C., Acar, A., Uluagac, S., Montanari, R., Bellavista, P., Conti, M.: A Survey on Decentralized Identifiers and Verifiable Credentials. *IEEE Commun. Surv. Tutor.* (2025)
39. Mühle, A., Assaf, K., Meinel, C.: One to Bind Them: Binding Verifiable Credentials to User Attributes. In: *SECRYPT*. pp. 345–352 (2023)
40. Page, M.J., et al.: PRISMA 2020 explanation and elaboration: Updated guidance and exemplars for reporting systematic reviews. *BMJ* **372** (2021)
41. Park, H.A.: The DtMin Protocol: Implementing Data Minimization Principles in Medical Information Sharing. *Electronics* **14**(8) (2025)
42. Pu, Y., Chen, J., Pan, J., Yan, D., Zhang, X., Ji, S.: Facial Data Minimization: Shallow Model as Your Privacy Filter. *IEEE Trans. Dependable Secure Comput.* (2025)

43. Rahul, K., Banyal, R.K.: Data Life Cycle Management in Big Data Analytics. In: *Procedia Comput. Sci.* vol. 173, pp. 364–371 (2020)
44. Ramić, Š.B., Cogo, E., Prazina, I., Cogo, E., Turkanović, M., Mulahasanović, R.T., Mrdović, S.: Selective disclosure in digital credentials: A review. *ICT Express* **10**(4), 916–934 (2024)
45. Ramić, S.B., Prazina, I., Pozderac, D., Mulahasanović, R.T., Mrdović, S.: BLS-MT-ZKP: A Novel Approach to Selective Disclosure of Claims from Digital Credentials. *IEEE Access* **12**, 192062–192078 (2024)
46. Rastegarpanah, B., Gummadi, K.P., Crovella, M.: Auditing Black-Box Prediction Models for Data Minimization Compliance. In: *NeurIPS* (2021)
47. Russo, A., Lax, G., Dromard, B., Mezred, M.: A System to Access Online Services with Minimal Personal Information Disclosure. *Inf. Syst. Front.* **24**(5), 1563–1575 (2022)
48. Schardong, F., Custódio, R.: Self-Sovereign Identity: A Systematic Review, Mapping and Taxonomy. *Sensors* **22**(15) (2022)
49. Shanmugam, D., Diaz, F., Shabanian, S., Finck, M., Biega, A.: Learning to Limit Data Collection via Scaling Laws: A Computational Interpretation for the Legal Principle of Data Minimization. In: *ACM Int. Conf. Proc. Ser.* pp. 839–849 (2022)
50. Staab, R., Jovanović, N., Balunović, M., Vechev, M.: From Principle to Practice: Vertical Data Minimization for Machine Learning. In: *IEEE S&P.* pp. 4733–4752 (2024)
51. Staab, R., Jovanović, N., Mai, K., Ganesh, P., Vechev, M., Fioretto, F., et al.: SoK: Data Minimization in Machine Learning (2025)
52. Tran, C., Fioretto, F.: Data minimization at inference time. *Advances in Neural Information Processing Systems* **36**, 72248–72269 (2023)
53. W3C: Verifiable Credentials Use Cases (2019), <https://www.w3.org/TR/vc-use-cases/>
54. W3C: Decentralized Identifiers (DIDs) v1.1 (2025), <https://www.w3.org/TR/did-1.1/>
55. W3C: Verifiable Credentials Data Model v2.0 (2025), <https://www.w3.org/TR/vc-data-model-2.0/>
56. Watanabe, H., Uchikoshi, M.: Generating Privacy-Preserving Personalized Advice with Zero-Knowledge Proofs and LLMs. In: *Web Conf. Companion* (2025)
57. Yan, L., Ma, M., Li, D., Huang, X., Ma, Y., Xie, K.: A flexible and lightweight privacy-preserving handshake protocol based on DTLShps for IoT. *Comput. Netw.* **216** (2022)
58. Yang, Q., Lepore, C., Eynard, J., Laborde, R.: From theory to practice: Data minimisation and technical review of verifiable credentials under the gdpr. *Computer Law & Security Review* **57**, 106138 (2025)
59. Yin, H., Chen, S., Luo, Z., Tang, M., Zhang, S., Yao, Z.: A Blockchain-Based Selective Disclosure Authentication System: A Self-sovereign Credential Scheme Combining Decentralized Identity and Zero-Knowledge Proofs. In: *CCIS.* vol. 2469, pp. 431–441 (2025)
60. Zhang, J., Wang, X., Cui, J., Li, R., Zhong, H.: A Decentralized Threshold Credential Management With Fine-Grained Authentication for VANETs. *IEEE Trans. Inf. Forensics Secur.* **20**, 5818–5831 (2025)
61. Zhong, J., Abreu, T., Souihi, S., Lucas, F.S.: G-TOK: Zero-Knowledge Proofs Based Dynamic Verifiable Credentials for Sensor Data Sharing. In: *BCCA.* pp. 400–405 (2024)
62. Zhou, J., Xu, E., Wu, Y., Li, T.: Rescriber: Smaller-LLM-Powered User-Led Data Minimization for LLM-Based Chatbots. In: *CHI* (2025)

Regulatorily Aligned Consult-Scoped Anonymous Credentials with Accountable Transparency for EHDS-based Health Data Access

Mahdi Akil^[0000–0002–7863–2281] and Frank Pallas^[0000–0002–5543–0265]

University of Salzburg, Salzburg, Austria
{mahdi.akil, frank.pallas}@plus.ac.at

Abstract. Inter-hospital consultations require a specialist to see enough of a patient’s record to answer a concrete clinical question, yet current sharing mechanisms often disclose more than the question requires. The European Health Data Space (EHDS) adds a further constraint: patients must be able to learn who accessed which categories of their data, and when. Full anonymity of the accessing clinician is therefore incompatible with the required transparency functionality, just as cleartext identification is unnecessary for routine scope enforcement and should be avoided under GDPR data minimisation. We present a consult-scoped delegation protocol designed from the outset around the resulting design point: pseudonymous access with auditor-controlled resolution. A treating physician obtains a case-bound policy token, then delegates a fresh credential to the consulted specialist, who proves possession to the hosting gateway while disclosing only the scope values needed for enforcement. Each access is logged under an encrypted identity that an auditor can resolve, aligning the protocol explicitly with the EHDS transparency obligation it serves.

Keywords: anonymous credentials, accountable transparency, delegation, EHDS, health data access

1 Introduction

When a treating physician Dr. A at hospital H1 consults a specialist Dr. B at hospital H2, existing record-sharing workflows are often coarse grained relative to the consultation question. In pull-style models, Dr. B commonly accesses external information through a portal to retrieve documents from H1 [7,19]. In push-style/document-exchange workflows, H1 may generate and send a standardised clinical summary, such as a Continuity of Care Document (CCD) or Consolidated CDA (C-CDA), to H2 [7,14]. These patterns are clinically useful, but the accessible content is largely shaped by portal/document organisation, exchange policy, and C-CDA templates rather than the specific consultation question. As a result, they may require clinicians to review information beyond what is needed and may contribute to information overload [7,14,19].

The European regulatory context sharpens this problem. The General Data Protection Regulation (GDPR) requires data minimisation, purpose limitation,

and transparency, among other principles [10,16]. Where the GDPR states these principles in general terms, the European Health Data Space (EHDS) Regulation makes access transparency concrete for the health domain. For primary use, that is the delivery of care itself rather than secondary research or policy use, patients should be able to learn which healthcare provider or other individual accessed their data, when, and which personal electronic health data were accessed [11]. This is what makes the setting specific to the EHDS rather than a generic external-sharing problem such as supply-chain disclosure. This creates a design tension between minimal disclosure during authorisation and accountable transparency after access: the gateway should enforce the case, categories, and expiry without learning the accessor’s professional identity, and no more about the delegator than the short-lived authorisation token it must verify, while the access log must still carry a pseudonymous entry an authorised auditor can resolve when transparency policy requires. The central question is how to combine minimal disclosure, consult-specific delegation, and resolvable accountability in one workflow.

We answer this question with a protocol that combines delegatable anonymous credentials (DACs) [3,4], verifiable encryption [6], and non-interactive zero-knowledge proofs [12]. Unlike generic anonymous-credential and delegation schemes, which optimise for unlinkability alone, our design is shaped throughout by the EHDS requirement that each access remain resolvable to the clinician who performed it. Full anonymity of the consulted specialist is thus not the design goal: Art. 9 EHDS requires every access to remain resolvable to a natural person, while GDPR data minimisation favours avoiding unnecessary cleartext identification at access time. Pseudonymity with auditor-controlled resolution is therefore the design point the two regulations strongly motivate, and the protocol is built around it from the outset rather than around anonymity subsequently weakened for compliance. The treating physician obtains a case-bound authorisation from her hospital, then delegates a fresh consult-scoped credential to the specialist. At access time the specialist proves possession to the hosting gateway while revealing only the scope values needed for enforcement, and the access is logged under an encrypted identity that an auditor can resolve for EHDS transparency.

The contribution is therefore not a new anonymous-credential primitive in isolation, but an EHDS-specific protocol composition for primary-use health-data access. Delegation is tied to a concrete clinical case, purpose, scope, and expiry, the gateway learns only the values it must enforce, and the resulting access log remains resolvable to the clinician who actually performed the access. This combination distinguishes the protocol from generic anonymous-credential delegation, where unlinkability is typically the main design goal, and from encryption-based health-record sharing, where the central problem is data release rather than accountable live access. More broadly, it follows the view that PETs in the EHDS should operationalise legal and governance requirements rather than stand apart from them [9].

The remainder of the paper is organised as follows. Section 2 reviews the EHDS transparency obligation and cryptographic primitives. Section 3 describes the participants, the threat model, and derives the security and privacy requirements from the EHDS transparency obligation. Section 4 specifies the protocol, and Section 5 argues that it meets the requirements. Section 6 reports indicative performance figures. Section 7 discusses related work, Section 8 covers deployment and the patient-transparency policy question, and Section 9 concludes the paper.

2 Background and Design Ingredients

In this section we set out the two ingredients the protocol is built from, the EHDS transparency obligation that motivates its design, and the cryptographic building blocks it relies on. We keep the explanation informal and refer to the original works for formal definitions, focusing instead on the properties each primitive provides and how they fit together in the consult-scoped access path.

The European Health Data Space (EHDS). Regulation (EU) 2025/327 governs electronic health data in the EU and, for primary use, strengthens patient-facing access rights. Beyond a general right of access to one’s own health data (Art. 3), it gives patients the right to obtain information about access to their electronic health data: under Art. 9 this information must cover at least who accessed the data, when, and which data were accessed, and must be retained for at least three years. Recital 16 reinforces that access by providers or other individuals should be transparent to the patient. Pure anonymity is therefore not the right design point, but neither is full identification. An access event must remain attributable to the accessing clinician (the consulted specialist Dr. B) after the fact. At the same time, the GDPR data-minimisation principle requires that at access time the gateway and any network observer learn no more than is needed to enforce the consult, so the accessor’s professional identity should not be exposed in the clear. Taken together, the two obligations point to a narrow design point: pseudonymisation in the sense of GDPR Art. 4(5), where attribution is impossible without additional information held separately under controls. This is the technical meaning we assign to the legal requirement, and it is why the protocol escrows an encrypted, resolvable identifier rather than either exposing or irreversibly hiding the accessor.

A zero-knowledge proof of knowledge lets a prover convince a verifier it knows secret witnesses satisfying a public relation while revealing nothing else [13]. In $ZPK\{(\alpha, \beta) : y = g^{\alpha}h^{\beta}\}$ the quantities in parentheses are the secret witnesses, and all other symbols are public. A *signature proof of knowledge* on message m , $SPoK\{\dots\}(m)$, is the non-interactive variant Fiat–Shamir [12], which binds m ’s integrity to the proof.

Attribute-based and delegatable credentials. ABCs let a user authenticate while disclosing only the needed attributes [8,5]. We write $C_I(a_0, \dots, a_L)$ for a credential issued by I to a user with private key $a_0 = k_U$ and attributes $a_1, \dots, a_L$. Presentation reveals a chosen subset A_d while hiding A_h (including

k_U), proved by $PK\{A_h : C_I(A_h; A_d)\}(m)$. *Delegatable* anonymous credentials (DACs) additionally let a holder pass a credential on [3,4]. We follow the construction of Camenisch et al. [4]. At a high level, issuer I (root) signs user A 's public key together with an attribute set to give A a Level-1 credential. A may then sign B 's public key together with a *delegated* attribute set a' (chosen by A) to give B a Level-2 credential, and B presents a non-interactive proof of possession of the chain that hides the public keys and non-disclosed attributes from third-party verifiers and requires only pk_I to verify.

Verifiable encryption [6] lets a party encrypt a value and, at the same time, prove that the ciphertext is well-formed and that the plaintext satisfies a stated relation, all without revealing the plaintext. We use it for identity escrow. The accessing doctor's identifier id_B is encrypted as $\psi \leftarrow \text{enc}(pk_{\text{Aud}}, id_B)$ under the auditor's public key pk_{Aud} , and the access proof carries a well-formedness relation showing that ψ encrypts the same id_B that appears in the delegatee's credential. Gateways and network observers see only ψ . The auditor can resolve it when policy requires patient-facing transparency. We use the term *pseudonymous* throughout in the sense of GDPR Art. 4(5): ψ cannot be attributed to a specific clinician without the additional information sk_{Aud} , which is kept separately by the auditor under its own controls. The design therefore relies on encryption rather than hashing, because the EHDS transparency right requires the link to be recoverable when policy demands, so the identifier must be pseudonymous and resolvable rather than anonymous.

3 System and Requirements

This section defines the protocol model. We describe the participants and the trust assumptions for each role, state the threat model, and identify the EHDS transparency obligation that motivates the design.

3.1 Participants and Trust Assumptions

Figure 1 shows the architecture. The six entities are described below:

- **Credential issuer CI.** Enrols licensed doctors, issuing each a Level-1 credential together with a matching identity certificate. In practice this role can be played by a national or regional health-professional authority.
- **Delegator and delegatee.** The treating physician (delegator) and the consulted specialist (delegatee) carry these credentials and run the consult between them. These are ordinary clinicians typically at different hospitals.
- **Access gateway GW.** Operated by each hospital, it verifies access proofs and logs accesses. It corresponds to the access-control front end of a hospital's EHR or health information exchange endpoint.
- **Policy decision point PDP.** Also hospital-operated, it authorises out-bound consultations for cases under the hospital's custody. This maps onto a hospital's existing case-management or authorisation engine, which already knows which clinician is responsible for which case.

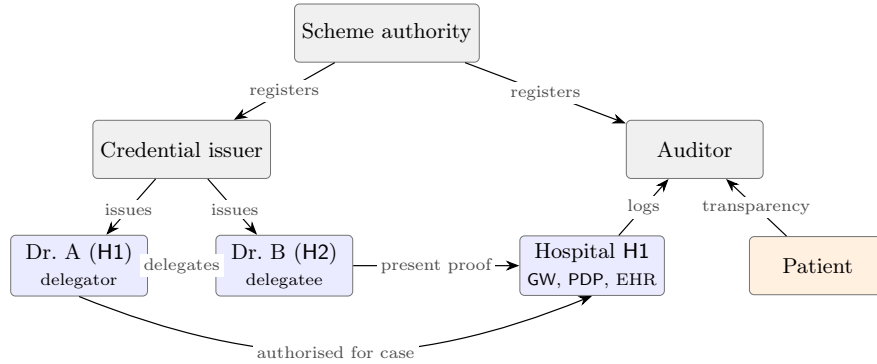


Fig. 1. System architecture for consult-scoped delegation and transparency. Dr. A (Hospital H1) is authorised by H1’s policy decision point for the case and delegates a consult-scoped credential to Dr. B (Hospital H2), who presents an access proof to H1’s gateway. Same-hospital consultations use the same flow with the local PDP and GW.

- **Auditor Aud.** The only authority able to resolve the pseudonymous access-log identifier back to the accessing doctor. A natural real-world home for this role is an EHDS access-information service or a data-protection authority that patients can later query.
- **Scheme authority SA.** Oversees the others: it registers the credential issuer, auditor, and hospitals and publishes the public parameters they share, so every party has one trusted reference point. This logical role may be co-located with CI under distinct keys.

Our trust assumptions follow the role each party plays. We assume SA is honest during setup and CI is honest during enrolment and revocation, meaning it assigns each doctor a consistent identifier across the credential and the certificate. Gateways and the auditor are honest-but-curious: GW enforces scope and expiry as specified but may try to infer hidden attributes from what it sees, and Aud resolves identifiers only under defined controls, though by design its key is the capability that makes recovery possible at all. Doctors, by contrast, may be corrupted. The adversary may observe the network, collude with a gateway, and obtain the long-term secret of at most one of CI and Aud after enrolment. A holder of sk_{Aud} can naturally decrypt routine logs. Section 8 discusses threshold deployment to reduce this concentration of trust.

3.2 Requirements

Table 1 summarises the requirements. Several follow directly from the regulatory tension of Section 2: every access must stay attributable after the fact (G7), yet at access time the gateway needs only the case, category, and validity window, so disclosing the accessor’s identity would reveal on every routine access which

Table 1. Security and privacy requirements.

Goal	Requirement
G1	Minimal disclosure: an access reveals nothing about the accessing specialist beyond the scope the gateway enforces.
G2	Scope and freshness: access is confined to the authorised case and data categories, is valid only within the consult’s lifetime, and a captured access cannot be reused later or redirected.
G3	Delegation integrity: only enrolled clinicians can take part, the specialist cannot be impersonated, and a consult received from a colleague cannot be passed on further.
G4	Case-bound authorisation: a consult is permitted only when the treating physician is genuinely responsible for the named case and the requested sharing fits the stated purpose. Holding a valid clinician credential is not by itself sufficient.
G5	Routine-log pseudonymity: ordinary access logs do not reveal the accessing doctor to the gateway or to network observers.
G6	Cross-consult unlinkability: separate consults by the same specialist cannot be linked to one another from what verifiers see.
G7	Patient transparency: a patient can later obtain a record of who accessed their data, and the accessor can, when policy requires, be resolved to the actual clinician who performed the access.

specialist treats which patient (G1, G5, G6). The consult workflow itself adds scope and freshness (G2), delegation integrity (G3), and case-bound authorisation (G4), the last requiring that the treating physician be genuinely responsible for the case and that the sharing match a stated purpose, so that a valid clinician credential is not on its own enough to open a consult. Section 5 returns to each requirement.

4 Protocol

This section specifies the protocol. We first fix notation and the setup that the scheme authority and credential issuer establish once. We then describe the consult itself in two parts: how a treating physician obtains a case-bound authorisation and delegates a fresh consult-scoped credential to the specialist, and how the specialist later presents that credential to the gateway, which checks it, enforces only the consult scope, and logs a pseudonymous entry that an auditor can later resolve for transparency.

4.1 Notation and Setup

Let C_d be doctor d ’s Level-1 credential, $C_{A \rightarrow B}$ a Level-2 credential delegated from Dr. A to Dr. B, and $scope_{AB} = (case_id, \mathcal{C}, t_{exp})$ the consult scope, where the permitted category set $\mathcal{C}$ is defined by exact-node or subtree selectors in

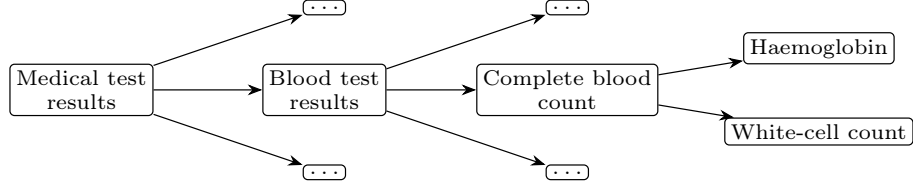


Fig. 2. Example category hierarchy, not a prescribed clinical vocabulary. Arrows go from broad to narrow, and ellipses indicate omitted sibling categories. A subtree grant covers all descendants. An exact-node grant covers only the named node.

a versioned hierarchy $\mathcal{H}_{\mathcal{C}}$ and stored as the scope hash $\langle \mathcal{C} \rangle = H(\text{version}, \mathcal{C})$. Figure 2 illustrates these two selector types. Credential attributes are position-bound, so an equality in a proof is checked against a specific attribute and cannot be satisfied by the right value sitting in the wrong field. The maximum delegation depth is fixed to two.

During setup, SA establishes the common parameters and public keys that all later proofs and verifications rely on. It publishes the bilinear-group parameters for the DAC system, hash functions H and H_{dom} , the auditor’s escrow public key pk_{Aud} , the issuer public keys pk_{CI} and pk_{CI_1} , and certified gateway and PDP keys. A certified trust registry supplies their identifiers, validity periods, and the signed hierarchy version. Parties reject expired keys and unknown versions. At enrolment, CI issues each doctor d a Level-1 credential

$$C_d = ((\sigma_{\text{CI},d}, a_d, pk_d), sk_d),$$

where $\sigma_{\text{CI},d}$ is the issuer’s signature binding the attributes a_d and the credential key pk_d , and sk_d is d ’s credential secret. CI also issues an identity certificate $Cert_d$ carrying the same identifier id_d as a_d , and the access proof relies on this consistency to bind the escrowed identifier to a certified clinician identity.

Design rationale. The protocol deliberately keeps three authorities apart, each answering a different question. The issuer attests that a credential belongs to a licensed doctor. The policy decision point attests that a particular physician is entitled to open a consult for a particular case. The auditor is the sole party able to recover an accessor’s identity from routine logs. Were any of these roles folded into the gateway, the gateway would either learn more than it needs at access time or become trusted for both enforcement and identity recovery, defeating the minimal-disclosure goal. Keeping them separate also lets a deployment swap one component without disturbing the rest: a national PKI can serve as CI, a hospital policy engine as PDP, and an EHDS access-information service as Aud.

4.2 Consult Delegation

A consult proceeds in three delegation phases, after which the specialist presents the result to a gateway (the access presentation, treated separately in Section 4.3):

1. **Policy authorisation:** Dr. A obtains a signed authorisation token $auth_A$ from H1's PDP, binding her to the case, scope, purpose, and validity window.
2. **Certificate exchange:** Dr. A and Dr. B exchange identity certificates, so that Dr. A learns the delegatee key pk_B and identifier id_B .
3. **Delegation:** Dr. A issues Dr. B a fresh Level-2 credential scoped to the consult and bound to $auth_A$.

Policy authorisation. Dr. A requests a token from H1's PDP before delegating:

$$req = (pk_A, case_id, \mathcal{C}, t_{exp}, purpose),$$

where *purpose* is a purpose-of-use drawn from a hospital vocabulary for example *specialist-consult* or *second-opinion*. The PDP checks against H1's registries that Dr. A is a clinician responsible for *case_id*, that the categories in $\mathcal{C}$ may be shared externally for the stated purpose, and that t_{exp} respects local policy. On success it generates a fresh nonce n and a short validity window $[not_before, not_after]$, and signs

$$body_A = (pk_A, case_id, \mathcal{C}, t_{exp}, purpose, n, not_before, not_after),$$

and returns $auth_A = (body_A, sig_A)$, where $sig_A = \text{sign}(sk_{PDP,1}, body_A)$. The token is short-lived: it authorises construction of the consult delegation, not open-ended access.

Certificate exchange. Dr. A and Dr. B exchange identity certificates over a mutually authenticated and confidential clinical channel. Dr. A verifies Dr. B's certificate under pk_{Cl_1} , extracts pk_B and id_B , and requires proof that Dr. B controls the matching signing key. A separate challenge-response is run only if the authenticated channel does not already provide equivalent key-possession assurance. This check establishes only that Dr. B is present at handshake time and does not by itself prevent the delegation from later being misused. That guarantee comes from the access proof instead: because presenting the credential requires proving knowledge of sk_B , only Dr. B can ever use the delegation, even if the credential material is intercepted or forwarded.

Delegation. Dr. A can now issue the Level-2 credential for Dr. B. To tie it to the specific authorisation $auth_A$ she obtained from PDP, she computes a *binding tag* $h_A = H_{dom}(auth_A)$, a hash of the PDP token, and places it inside the credential. She forms the Level-2 attributes

$$a' = (id_B, case_id, \langle \mathcal{C} \rangle, t_{exp}, h_A),$$

which records Dr. B's identifier, the case, the scope hash, the expiry, and the binding tag. She delegates to Dr. B by signing (pk_B, a') with her credential key, producing the delegation signature σ_A .

Dr. B receives $(\sigma_A, a', auth_A, \sigma_{Cl,A}, a_A, pk_A)$ and verifies that everything is consistent: the DAC chain is valid, the PDP signature on $auth_A$ verifies, the tag in the credential matches the token she was given ($a'[\text{tag}] = H_{dom}(auth_A)$), and the token body and a' agree on case, the scope hash, expiry, and delegating key. On success Dr. B stores the consult credential

$$C_{A \rightarrow B} = ((\sigma_{Cl,A}, a_A, pk_A), (\sigma_A, a', pk_B), sk_B).$$

The key property is that a later access is accepted only when all three hold together: a valid clinician credential, a delegation signature, and a matching case-bound PDP token. No one of them suffices on its own.

4.3 Access Proof

To access category c of case $case_id$ at gateway gw_id , Dr. B chooses a timestamp ts , sets $m = case_id \parallel c \parallel ts \parallel gw_id$ and $\ell = H(\text{"EHDS-access"} \parallel m)$, and computes a fresh-labelled escrow ciphertext $\psi = \text{enc}(pk_{\text{Aud}}, id_B; r, \ell)$. She then produces the following signature proof of knowledge:

$$\begin{aligned} \pi = \text{SPoK}\{ & (\sigma_{\text{Cl},B}, \sigma_{\text{Cl},A}, \sigma_A, sk_B, pk_B, a_B, a_A, a', id_B, r) : \\ & C_B = ((\sigma_{\text{Cl},B}, a_B, pk_B), sk_B) \wedge a_B[id] = id_B \\ & \wedge C_{A \rightarrow B} = ((\sigma_{\text{Cl},A}, a_A, pk_A), (\sigma_A, a', pk_B), sk_B) \\ & \wedge a'[id] = id_B \wedge a'[\text{tag}] = H_{\text{dom}}(auth_A) \\ & \wedge a'[\text{case}] = case_id \wedge a'[\text{cat}] = \langle C \rangle \wedge a'[\text{exp}] = t_{\text{exp}} \\ & \wedge \psi = \text{enc}(pk_{\text{Aud}}, id_B; r, \ell) \} (m, auth_A, \psi). \end{aligned}$$

The proof establishes four things:

- Dr. B holds her own Cl-issued Level-1 credential, and the identifier inside it is id_B .
- She holds a Level-2 delegation whose root is the delegating key named in $auth_A$, so the delegation belongs to the case the PDP authorised rather than to some other token.
- The case, category, and expiry carried in the delegation equal the values Dr. B discloses to the gateway, so the scope the gateway enforces is exactly the scope signed into the credential and cannot be widened.
- The escrow ciphertext ψ encrypts that same id_B .

The proof shows that whoever produced this proof is a legitimately enrolled clinician, acting under a genuine case authorisation, within the scope that authorisation allows, and is the very person whose identity is sealed in ψ for later transparency. Algorithm 1 shows how the gateway checks this proof, performing the cleartext scope and freshness tests before verifying π itself.

Algorithm 1 Gateway verification, abbreviated

Require: $(case_id, c, C, t_{\text{exp}}, ts, auth_A, \psi, \pi)$ and public keys

- 1: reject unless $|now - ts| \leq \delta_t$, $c \in \text{Allow}_{\mathcal{H}_C}(C)$, and $ts < t_{\text{exp}}$
 - 2: reject if $H(\pi \parallel \psi \parallel case_id \parallel c \parallel ts)$ is already in the replay cache
 - 3: parse $auth_A = (body_A, sig_A)$ and reject unless the token scope matches the request and its validity window contains ts
 - 4: reject unless $\text{verify}(pk_{\text{PDP},1}, body_A, sig_A) = \top$
 - 5: set gw_id to this gateway's certified local identifier and recompute m ; verify π against $pk_{\text{Cl}}, pk_{\text{Aud}}, pk_A, auth_A, m, \psi$
 - 6: check local case status; cache the replay key; log; release $D[case_id, c]$ through the authenticated response channel
-

In this pull flow, GW queries H1’s EHR and returns matching resources. A FHIR adapter maps selectors to FHIR types, profiles, codes, or labels and writes an `AuditEvent`.

4.4 Logging, Transparency, and Revocation

Logging. For every granted access, GW records a log entry

$$L = (case_id, c, ts, \psi, \sigma_{GW}),$$

where σ_{GW} is the gateway’s signature over the preceding fields. The entry is stored locally and forwarded to `Aud`, indexed by a patient correlation identifier so that the auditor can later retrieve all accesses to a given patient’s record. A hash chain or Merkle log and signed checkpoints replicated to `Aud` detect deletion, rollback, and alteration without a blockchain. These fields line up with what EHDS Art. 9 requires of an access record: the category c records *which data* were accessed, ts records *when*, and ψ carries *who*, in encrypted form. The log uses the most specific released hierarchy code, not only a broad parent. The purpose checked at authorisation stays inside $auth_A$ and is deliberately not copied into the log, since Art. 9 does not require it and a purpose value can itself reveal sensitive context such as the specialty involved; a deployment that wants purpose in the patient-facing record can surface it as a policy choice. Only the delegatee is recorded as the accessor, sealed in ψ .

Transparency. When a patient requests access information, `Aud` authenticates the patient out-of-band, retrieves the relevant entries, and returns, for each entry, the tuple $(case_id, c, ts, accessor)$. The form of *accessor* is a policy choice: a pseudonymous reference, a role or institution summary, or the fully resolved identifier $id_B = \text{dec}(sk_{\text{Aud}}, \psi)$. Whichever form is used, the resolution is faithful. Because the access proof binds ψ to the identifier in Dr. B’s own credential, a resolved entry names the doctor who actually accessed the record and never Dr. A, even though Dr. A created the delegation. Section 8 discusses which of these forms a deployment should expose by default.

Revocation. Revocation operates at the two credential levels independently. A doctor’s Level-1 credential is revoked through the issuer’s standard mechanism, for example a revocation accumulator or short-lived validity attributes, which GW checks at access time. A Level-2 consult credential normally expires on its own at t_{exp} , so no explicit revocation is needed in the common case. For exceptional early revocation, a short revocation token for $scope_{AB}$ can be pushed to the gateway, which then refuses further accesses under that scope. Folding non-revocation witnesses directly into π is compatible with the underlying DAC construction [4], but we treat it as a deployment extension rather than part of the core protocol. Revocation blocks later access but cannot undo disclosure. Its log remains evidence.

5 Security and Privacy Analysis

We argue informally that the protocol satisfies G1–G7 under the assumptions of Section 3 and the security of the DAC, verifiable-encryption, signature, and proof systems.

Minimal disclosure (G1). The access proof discloses only $case_id$, c , $\mathcal{C}$, t_{exp} , ts , $auth_A$, and ψ . It hides Dr. B’s Level-1 attributes apart from the equality-bound identifier, hides the non-disclosed attributes in Dr. A’s credential, and keeps the delegatee identifier from GW behind the escrow ciphertext. The gateway therefore learns only the scope information it needs to enforce the consult, and no other specialist attribute or cleartext identity. Section 8 notes that $auth_A$ exposes the delegator key and consult context.

Scope and freshness (G2). The scope values Dr. B hands the gateway in the clear are not just unverified claims. Soundness of the SPoK binds them to the signed Level-2 attributes $a'[case]$, $a'[cat]$, and $a'[exp]$, so an adversary cannot substitute a broader case, category set, or expiry without producing a fresh proof. The gateway runs the checks $c \in \text{Allow}_{\mathcal{H}_C}(\mathcal{C})$ and $ts < t_{exp}$ in the clear (Algorithm 1), but they apply to values cryptographically tied to the delegated credential. The fresh timestamp, the gateway identifier inside m , and the replay cache together ensure that a captured proof cannot be replayed later or redirected to another gateway. Each verifier uses its certified gw_id in m , preventing redirection. Only replicas of one logical gateway share a replay cache.

Delegation integrity (G3). Unforgeability of the DAC scheme implies that an accepted chain must originate in a CI-issued Level-1 credential and contain a valid second link signed by the delegator’s credential key [3,4]. Non-transferability follows because presentation requires sk_B : lending the delegation would mean lending the delegatee’s own professional credential secret. The setup fixes the maximum depth to two, so any attempt to re-delegate further fails the verification predicate.

Case-bound authorisation (G4). This is the main difference between a generic delegation mechanism and a consult-safe one. The gateway accepts a proof only if the disclosed PDP token verifies, the token body matches the visible request, the Level-2 tag equals $H_{dom}(auth_A)$, and the proof fixes the first-link key of the DAC chain to pk_A , the key named in the token. A clinician therefore cannot pair a delegation with a token issued to a different key, and a valid clinician credential on its own is not enough to delegate a case. The guarantee still depends on an honest PDP registry, since a compromised PDP can authorise according to a false view, but the protocol is strictly stronger than role-only delegation because it requires explicit case binding.

Routine-log pseudonymity (G5). Ordinary log entries carry ψ in place of the doctor’s identifier. Under the privacy of the verifiable-encryption scheme [6], a gateway or network observer without sk_{Aud} cannot distinguish encryptions of candidate identifiers, so routine logging reveals nothing about which doctor performed the access.

Cross-consult unlinkability (G6). Each access uses fresh encryption randomness under an access-context label, so two consults by the same specialist

produce unrelated ciphertexts. An honest verifier sees no stable per-doctor handle across consults. The auditor sits intentionally outside this privacy boundary, as it must be able to resolve identifiers for transparency, and threshold or dual-control decryption can reduce but not remove that residual trust.

Patient transparency and faithful resolution (G7). The auditor can recover $id_B = \text{dec}(sk_{\text{Aud}}, \psi)$ and return a patient-visible access record, satisfying the transparency requirement. Crucially, the resolution is faithful even against a malicious delegator. A delegator might try to place an innocent doctor’s identifier in $a'[\text{id}]$ so that the log resolves to the wrong accessor, but the proof forces a single witness id_B to satisfy three relations at once: $a_B[\text{id}] = id_B$ in Dr. B’s own CI credential, $a'[\text{id}] = id_B$ in the delegated credential, and $\psi = \text{enc}(pk_{\text{Aud}}, id_B; r, \ell)$. If Dr. A writes any identifier other than Dr. B’s certified one, Dr. B cannot produce a verifying proof. A resolved entry therefore names the doctor who actually accessed the record, never Dr. A. This threat is specific to a delegator-supplied identifier and does not arise in escrow schemes where the accessor encrypts its own identity, such as the inspector-based VANET constructions [1,2].

6 Performance

This section evaluates the cost of the consult access path. We measure each cryptographic primitive, combine the measurements with the operation counts of proof generation and verification, and report the resulting access-proof-stage timings.

We instantiate the pairing group with BLS12-381 and the Camenisch–Shoup identity-escrow group with a 3072-bit modulus, matching the curve’s ≈ 128 -bit security level¹. All costs are measured² on an Apple M4 laptop inside the `python:3.10-slim` Docker image (Linux ARM64, kernel 6.12 aarch64), with the pairing group provided by `petrelc/RELIC`, the escrow group by `GMP 6.3.0` via `gmpy2`, and hashing by `SHA-256`. We size the access path with attribute-vector lengths $k_1=5$ and $k_2=5$ at the two credential levels, $w=k_1+k_2+3=13$ hidden witnesses, four disclosed scope attributes, and fixed delegation depth $L=2$. We report unit costs and phase totals as the mean and sample standard deviation, using 200 samples for unit costs and $R = 1000$ runs for phase totals.

Table 2 reports the measured costs. The measured cryptographic stage totals about 144 ms, with proof generation at ≈ 83 ms and gateway verification at ≈ 61 ms. Complete authorisation, delegation, network, lookup, and EHR-retrieval stages await the full implementation and are not measured. The additional protocol-specific overhead of these stages is expected to be limited because authorisation and delegation are performed once per consult rather than for every subsequent access, while network, lookup, and EHR retrieval are deployment-dependent costs that also arise in conventional remote EHR access. This cost

¹ Following the comparable-strength guidance of NIST SP 800-57 Part 1 Rev. 5, <https://csrc.nist.gov/pubs/sp/800/57/pt1/r5/final>.

² The operation-count measurement script is available at <https://github.com/mahdi-akil/consult-scoped-ac-benchmark>.

Table 2. Per-operation unit cost (mean and sample standard deviation) and the number of times each operation is used in proof generation and verification, with the totals.

Operation	Unit (ms)	Generate	Verify
$\mathbb{G}_1$ exponentiation	0.056 ± 0.002	20	19
$\mathbb{G}_2$ exponentiation	0.170 ± 0.001	3	0
$\mathbb{G}_T$ exponentiation	0.293 ± 0.003	13	13
Pairing	0.765 ± 0.081	0	6
Exponentiation modulo n^2	12.89 ± 0.08	6	4
Hash	≈ 0	4	4
Total (ms)		82.7 ± 0.2	61.0 ± 0.1

is dominated not by the credential machinery but by identity escrow. The exponentiations modulo n^2 of the verifiable encryption account for roughly 77 of the 83ms of generation and 52 of the 61ms of verification, whereas the entire three-signature credential presentation costs only about 5 and 10 ms. Cost grows linearly in the signed-attribute counts k_1 , k_2 , and w , while the fixed depth $L=2$ keeps the chain short. Escrow is therefore the natural optimisation target. It could be moved into the credentials’ prime-order group using a discrete-log verifiable-encryption scheme such as that of Takahashi and Zaverucha [21], which supports efficient decryption of the escrowed identifier without the separate mod n^2 arithmetic and without touching the credential layer. A full end-to-end implementation integrated with FHIR/EHR infrastructure remains future work.

7 Related Work

Our protocol draws on two lines of healthcare work, namely access control for collaborative e-health and cross-domain encryption-based record sharing.

The closest line of work concerns access control for collaborative e-health, in which clinicians across administrative domains reach a record held elsewhere. Pussewalage and Oleshchuk develop this extensively, from attribute-based access with controlled delegation [17] to, closest to us, an anonymous delegatable attribute-based credential scheme with optional anonymity revocation [18]. Anonymous role-based schemes pursue similar goals while keeping accessors traceable [22]. These works establish the requirements we adopt, and are our most direct comparison. They differ in two ways: First, they delegate *attributes or rights*, so a suitably attributed credential suffices to act, whereas our delegation is bound to a specific case authorisation, so a valid clinician credential alone never opens a consult (G4). Second, all three predate the EHDS Art. 9 obligation and so do not target the patient-facing transparency it now requires (G7).

A parallel healthcare line shares records across organisational or cloud boundaries by encrypting them under attribute policies. Single-authority attribute-based encryption was applied to e-health records early on [15], and later work

Table 3. Comparison with the selected related healthcare schemes across the properties our setting requires. $\checkmark$ denotes the property is provided, $\times$ that it is not, and $\sim$ partial support. *Disclosure*, minimal disclosure at access time. *Deleg.*, controlled delegation. *Unlink.*, cross-access unlinkability. *Account.*, accountable de-anonymisation. *Case-bound*, authorisation tied to a specific case rather than to attributes alone. *Transp.*, resolvable patient-facing transparency, the capability required by EHDS Art. 9.

Scheme	Disclosure	Deleg.	Unlink.	Account.	Case-bound	Transp.
[17]	$\checkmark$	$\checkmark$	$\times$	$\times$	$\times$	$\times$
[18]	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\times$	$\times$
[22]	$\checkmark$	$\sim$	$\checkmark$	$\times$	$\times$	$\times$
[15]	$\checkmark$	$\times$	$\times$	$\times$	$\times$	$\times$
[20]	$\checkmark$	$\times$	$\sim$	$\times$	$\times$	$\times$
Ours	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$	$\checkmark$

moved to multi-authority and hierarchical constructions for cross-domain sharing [20]. These schemes control who may *decrypt* stored data, which is complementary to our problem. We do not encrypt the record under a policy but instead govern a live, case-scoped access at a hosting gateway while keeping the accessor pseudonymous, so encryption-based sharing and our consult-scoped delegation address different points in the inter-hospital workflow.

Table 3 positions our protocol against the relevant healthcare schemes along the properties our setting requires. The two right-most properties, case-bound authorisation and resolvable patient-facing transparency, are the ones that distinguish a generic delegation mechanism from a consult-safe one under the EHDS, and to our knowledge no prior healthcare scheme provides both.

8 Discussion

What should the patient see? The cryptographic mechanism deliberately leaves the form of the patient-visible accessor to policy. A strong-transparency setting decrypts ψ for every patient request and returns the doctor’s identifier. A strong-minimisation setting returns only a pseudonymous handle unless the patient asks for it. A practical middle ground is tiered transparency, where the default view reports the time, category, and case context together with a role or institution summary, and a separately logged resolution step reveals the doctor’s identifier only when the patient, a regulator, or local policy requires it. Our protocol design supports all three choices. We prefer the tiered design as a default since it preserves a meaningful patient-facing log while making identity resolution an auditable event. It also gives Member States room to interpret EHDS transparency consistently with local professional-confidentiality rules.

Deployment choices. The single-auditor model is simple but politically sensitive, since one party holds the capability to resolve every access. A deploy-

ment can instead split sk_{Aud} across, for example, a national authority, a professional body, and a data-protection office using threshold decryption, so that identity resolution requires several independent approvals. The auditor key can also be rotated periodically. The threshold shares for an old key can be retained for as long as access logs created under that key need to remain resolvable. As long as the required number of shares remains available, the auditor can still recover the encrypted identities in those logs. Compromise of an old key exposes the identities encrypted under that key. Multiple credential issuers can be supported by certifying issuer keys under the scheme authority and including issuer identity in the proof policy. A further choice is where to maintain the patient correlation identifier the auditor uses to retrieve entries.

Residual disclosure. The gateway sees $auth_A$, whose body contains the delegating credential key pk_A , the case scope, the purpose, and the validity window. This is a conscious trade-off: verifying the PDP token in the clear avoids the cost of proving its signature inside the zero-knowledge layer. The key is not a real-world name unless the gateway also holds a name-to-key mapping, but repeated use of the same short-lived token can correlate the accesses within a single consult. A deployment needing stronger delegator privacy could replace the cleartext token with a zero-knowledge proof that a valid authorisation was signed, at higher cost.

Limitations. The paper gives a design and an informal analysis, not a full UC proof or a production implementation. Formal proof and a complete FHIR/EHR prototype remain future work. The PDP is trusted for case assignment, and the auditor remains an empowered recovery authority unless threshold controls are deployed. Finally, the protocol targets primary-use consult access. Emergency “break-glass” access is outside the consult flow and must be separately authenticated and logged. Revocation cannot undo prior disclosure, and applicable law, including GDPR Art. 17 where applicable, governs log retention or erasure. Secondary use, lawful basis, consent capture, and patient-identity federation are surrounding governance problems rather than cryptographic sub-routines the protocol solves on its own.

9 Conclusion

Inter-hospital consultations need finer-grained sharing than today’s portal and document workflows provide, yet EHDS transparency also requires every access to remain attributable after the fact. We presented a consult-scoped delegation protocol combining delegatable anonymous credentials, case-bound PDP authorisation, zero-knowledge presentation, and verifiable identity escrow. The design minimises routine disclosure at the access gateway while preserving a patient-visible audit path through an auditor. Its two central technical points are the binding of each delegation to an explicit case authorisation, so that role membership alone cannot open a consult, and the triple binding of the accessor identifier across credential, delegation, and escrow ciphertext, which prevents a delegator from framing another doctor in the transparency log. Throughout, the

protocol is explicitly aligned to the EHDS provisions it serves: the resolvable escrow is derived from the Art. 9 transparency right rather than bolted onto an anonymity scheme, and minimal disclosure at the gateway operationalises GDPR data minimisation. We have given a design and an informal argument rather than a full proof or a deployed system, and both remain open, alongside a cheaper identity escrow and the governance question of how to split auditor trust. As the EHDS moves electronic health records toward patient-facing access transparency across the EU, the harder question is no longer whether minimal disclosure and accountable transparency can coexist, but how they should be balanced in practice, what a patient should see by default, and who decides.

Acknowledgments. This publication has been supported by the Excellence in Digital Sciences and Interdisciplinary Technologies (EXDIGIT) project, funded by Land Salzburg under the grant number 20204-WISS/263/6-6022.

References

1. Akil, M., Martucci, L., Hoepman, J.H.: Non-interactive privacy-preserving sybil-free authentication scheme in VANETs. In: Inaugural International Symposium on Vehicle Security and Privacy, VehicleSec 2023, San Diego, CA, USA. Internet Society (2023). <https://doi.org/10.14722/VEHICLESEC.2023.23007>
2. Akil, M., Martucci, L.A., Hoepman, J.: A secure and privacy-preserving authentication scheme with a zero-trust approach to vehicle renting in vanets. In: Proceedings of the 21st International Conference on Security and Cryptography, SE-CRYPT 2024, Dijon, France, July 8-10, 2024. pp. 114–127. SCITEPRESS (2024). <https://doi.org/10.5220/0012759600003767>
3. Belenkiy, M., Camenisch, J., Chase, M., Kohlweiss, M., Lysyanskaya, A., Shacham, H.: Randomizable proofs and delegatable anonymous credentials. In: Advances in Cryptology – CRYPTO 2009. LNCS, vol. 5677, pp. 108–125. Springer (2009). https://doi.org/10.1007/978-3-642-03356-8_7
4. Camenisch, J., Drijvers, M., Dubovitskaya, M.: Practical UC-secure delegatable credentials with attributes and their application to blockchain. In: Proc. 2017 ACM SIGSAC Conf. on Computer and Communications Security (CCS). pp. 683–699 (2017). <https://doi.org/10.1145/3133956.3134025>
5. Camenisch, J., Lysyanskaya, A.: An efficient system for non-transferable anonymous credentials with optional anonymity revocation. In: EUROCRYPT 2001. LNCS, vol. 2045, pp. 93–118. Springer (2001)
6. Camenisch, J., Shoup, V.: Practical verifiable encryption and decryption of discrete logarithms. In: Advances in Cryptology – CRYPTO 2003. LNCS, vol. 2729, pp. 126–144. Springer (2003). https://doi.org/10.1007/978-3-540-45146-4_8
7. Carequality and CommonWell Health Work Group: Concise consolidated CDA: Deploying encounter summary and patient summary documents with clinical notes. Tech. rep., Carequality (March 2022), Online report
8. Chaum, D.: Security without identification: Transaction systems to make big brother obsolete. Communications of the ACM **28**(10), 1030–1044 (1985). <https://doi.org/10.1145/4372.4373>
9. van Drumpt, S., et al.: Secondary use under the european health data space: setting the scene and towards a research agenda on privacy-enhancing technologies.

- Frontiers in Digital Health **7**, 1602101 (2025). <https://doi.org/10.3389/fdgth.2025.1602101>
10. European Union: Regulation (EU) 2016/679 (General Data Protection Regulation). Official Journal of the European Union, EUR-Lex (2016)
 11. European Union: Regulation (EU) 2025/327 on the European Health Data Space. Official Journal of the European Union, EUR-Lex (2025), see in particular Art. 3 (right of access), Art. 9 (right to information about data access)
 12. Fiat, A., Shamir, A.: How to prove yourself: Practical solutions to identification and signature problems. In: Conference on the theory and application of cryptographic techniques. pp. 186–194. Springer (1986). https://doi.org/10.1007/3-540-47721-7_12
 13. Fiege, U., Fiat, A., Shamir, A.: Zero knowledge proofs of identity. In: Proceedings of the nineteenth annual ACM symposium on Theory of computing. pp. 210–217 (1987). <https://doi.org/10.1007/BF02351717>
 14. Hosseini, M., Jones, J., Faiola, A., Vreeman, D.J., Wu, H., Dixon, B.E.: Reconciling disparate information in continuity of care documents: Piloting a system to consolidate structured clinical documents. Journal of Biomedical Informatics **74**, 123–129 (2017). <https://doi.org/10.1016/j.jbi.2017.09.001>
 15. Michalas, A., Weingarten, N.: HealthShare: Using attribute-based encryption for secure data sharing between multiple clouds. In: Proc. 30th IEEE Int. Symp. on Computer-Based Medical Systems (CBMS). pp. 811–815 (2017). <https://doi.org/10.1109/CBMS.2017.30>
 16. Pallas, F., Koerner, K., Barberá, I., Hoepman, J., Jensen, M., Narla, N.R., Samarin, N., Ulbricht, M., Wagner, I., Wuyts, K., Zimmermann, C.: Privacy engineering from principles to practice: A roadmap. IEEE Secur. Priv. **22**(2), 86–92 (2024). <https://doi.org/10.1109/MSEC.2024.3363829>
 17. Pussewalage, H.S.G., Oleshchuk, V.A.: Attribute based access control scheme with controlled access delegation for collaborative E-health environments. Journal of Information Security and Applications **37**, 50–64 (2017). <https://doi.org/10.1016/j.jisa.2017.10.004>
 18. Pussewalage, H.S.G., Oleshchuk, V.A.: An anonymous delegatable attribute-based credential scheme for a collaborative e-health environment. ACM Trans. Internet Techn. **19**(3), 41:1–41:22 (2019). <https://doi.org/10.1145/3338854>
 19. Rivera, R.L., et al.: Directly integrating health information exchange (HIE) data with the electronic health record increases HIE use by emergency department clinicians. ACI Open **7**(2), e49–e60 (2023). <https://doi.org/10.1055/s-0043-1772583>
 20. Roy, S., Agrawal, J., Kumar, A., Rao, U.P.: MH-ABE: Multi-authority and hierarchical attribute based encryption scheme for secure electronic health record sharing. Cluster Computing **27**, 6013–6038 (2024). <https://doi.org/10.1007/s10586-024-04283-z>
 21. Takahashi, A., Zaverucha, G.: Verifiable encryption from MPC-in-the-head. Cryptology ePrint Archive, Paper 2021/1704 (2021). <https://doi.org/10.62056/a3wa3z17s>
 22. Zhou, X., Liu, J., Liu, W., Wu, Q.: Anonymous role-based access control on e-health records. In: Proc. 11th ACM Asia Conf. on Computer and Communications Security (ASIACCS). p. 559–570. Association for Computing Machinery (2016). <https://doi.org/10.1145/2897845.2897871>

Designing and Verifying a Post-Quantum Protocol for Privacy Critical Applications using ABE and ProVerif

Robert-William Nii Odartei Evans¹[0009–0002–2007–0818] and Florian Kammüller^{1,2}[0000–0001–5839–5488]

¹ Middlesex University London, The Burroughs, Hendon, London, NW4 4BT

² Technische Universität Berlin, Straße des 17. Juni 135, 10623 Berlin, Germany

Abstract. Verifying cryptographic protocols provides mathematical assurance of their correctness. It enables system designers and engineers to place trust in their security and privacy guarantees prior to deployment and mass adoption. Given that security and privacy vulnerabilities are often expensive or impossible to remedy once systems are operational, formal verification is widely regarded as a prudent and necessary step in protocol development. This paper explores the integration of Covercrypt, an ETSI-standardised post-quantum attribute-based encryption (ABE) scheme, into a post-quantum secure protocol, by combining Covercrypt with post-quantum secure authentication and key-encapsulation. The protocol is designed to provide quantum-safe protection for privacy-critical data at rest while maintaining practical usability in enterprise environments. The proposed system targets cloud-based document storage and processing, where centralized policy enforcement, fine-grained access control for privacy, and long-term confidentiality are essential. Documents remain encrypted while stored yet can be accessed by authorized users. To assess the security and privacy of the protocol, several key properties are formally verified using ProVerif, a cryptographic model checker.

Keywords: Formal Verification · Post-quantum cryptography · Modelchecking.

1 Introduction

Recent advances in quantum computing are expected to fundamentally impact modern cryptographic systems. IBM predicts that by 2033 the full potential of quantum computers will be unlocked, enabling significant progress across industries such as healthcare, finance, and materials science. These advances pose a serious threat to classical public-key cryptographic schemes, particularly for data that must remain confidential over long periods of time. This has led to growing concern over the long-term security and privacy of cloud-hosted data and the need for cryptographic mechanisms that remain secure in a post-quantum setting. To aid in the transition from classical to quantum security, several forms

of encryption are being redesigned to protect against quantum capable attackers. One form of public-key encryption that needs post-quantum attention is Attribute-Based Encryption (ABE) [3]. ABE is a form of public-key encryption in which access to encrypted data is determined by a set of attributes possessed by a user, rather than by a specific user identity [18]. This approach contrasts with traditional role-based or identity-based access control models and allows for more expressive and fine-grained authorization policies. When combined with post-quantum cryptographic primitives, ABE offers a promising foundation for enforcing cryptographically verifiable access control while maintaining long-term data confidentiality.

Cryptographic primitives serve as the foundation of protocols which use them to secure communication over the Internet. The security capabilities of these protocols must be validated before widespread adoption. Formal methods play a critical role in the design and validation of secure systems like protocols. Model checking is a verification technique that explores all possible system states in a brute-force manner to determine whether specified properties hold [2]. Verifying cryptographic protocols provides mathematical assurance of their correctness and enables system designers and engineers to place trust in their security guarantees prior to deployment and mass adoption. Given that security and privacy vulnerabilities are often expensive or impossible to remedy once systems are operational, formal verification is widely regarded as a prudent and necessary step in protocol development. In 2025, a company called Cosmian released a new public key cryptographic scheme known as Covercrypt [4]. It promises post-quantum security alongside attribute-based encryption. In addition, this scheme has been standardized by ETSI [16]. Although a lot of work has been done with regard to post-quantum and ABE cryptographic primitives and schemes, application-layer protocols have been largely ignored by the community. Furthermore, there are very few formally verified post-quantum protocols which provide ABE.

To this end, this paper addresses the following research questions:

- RQ1: How can Covercrypt be integrated into a protocol to provide attribute-based access control while remaining resistant to quantum attacks?
- RQ2: Which security properties of such a protocol can be formally verified using ProVerif?

By combining only certified post-quantum secure cryptographic primitives of authentication and key exchange into our protocol, we provide a post-quantum secure protocol. This custom protocol will then be verified by the cryptographic model checker ProVerif, which uses the Dolev-Yao model[6]. After a short overview of Post-Quantum Cryptography in Section 2, we give an overview of our protocol together with key encapsulation algorithm ML-KEM and the digital signature algorithm ML-DSA as well as Covercrypt in Section 4. In the same section, we first introduce the ProVerif formalisation of some key concepts of post-quantum cryptography. Next, we present our protocol in the simple Alice-Bob notation to highlight the authentication steps using ML-DSA-signed nonces as well as the ML-KEM key distribution. In the remainder of Section 4, we present in detail the body of our work: a formal model of our custom protocol in ProVerif with a

detailed discussion of the verified security guarantees. These guarantees include mutual authentication, correctness of decryption, and confidentiality.

This paper is mainly based on the MSc Dissertation of one of the authors [8]. All ProVerif formalisations and implementations are available on GitHub [9].

2 Background

This section presents the foundational concepts underpinning the proposed protocol. We begin with an overview of a cryptographic verifier and its uses with regards to post-quantum protocols, followed by an examination of post-quantum cryptographic schemes which will be used in the protocol, and conclude with a discussion of attribute-based access control and its role in modern access control architectures.

2.1 ProVerif

ProVerif is a popular model checking tool that has been used to verify many communication protocols. It works on the Dolev-Yao formal model [7]. This model assumes that the attacker controls the network, meaning the attacker can eavesdrop, intercept, and construct their own messages based on previous messages seen. The second assumption is that cryptography is assumed to be perfect. The attacker cannot break any encryption unless they have a key that enables them to do so. This assumption can be difficult to translate into the real world; however, such assumptions serve effectively when reasoning about protocols symbolically. ProVerif was used to verify Signal’s post-quantum upgrade to its communication protocol [5]. This new protocol, known as Post-Quantum Extended Diffie-Hellman (PQXDH), provides post-quantum forward secrecy in addition to the other guarantees provided by the previous version of the protocol. To properly secure information systems against the quantum threat, all forms of encryption must be considered and upgraded.

2.2 Post-Quantum Schemes

Post-Quantum cryptography [14] has made several leaps over the past few years. In 1998, the computer scientist Ajtai [1] realised that mathematical structures known as lattices can be used as a foundation for new crypto-primitives. Another development in the post-quantum space was the Learning with Errors (LWE) problem, which was discovered by Oded Regev [17] in 2005. LWE was improved upon, and this led to new variants known as Ring-LWE and Module-LWE. Most existing post-quantum crypto primitives depend on Module-LWE, like ML-KEM formerly known as Kyber. It has been standardized by the National Institute for Standards and Technology (NIST) and work is currently being done to integrate it into popular communication protocols like TLS. As our communication systems are being upgraded with new and powerful algorithms, care must be taken to ensure that post-quantum protocols work as expected. To this end,

model checking can be seen as a useful tool to ensure the correctness of these upgraded protocols. Tools like ProVerif and Tamarin can be employed to ensure the correctness of these protocols.

ML-DSA Module-Lattice Based Digital Signature [15] is a post-quantum digital signature scheme standardized by NIST in 2024. The scheme has three main operations that enable it to function. The first is Key Generation, which creates a public and private key pair. The second operation is signing, which generates a signature using the private key and the third is verification, which checks the signature using the public key. The scheme also provides non-repudiation, which means that the signer cannot deny signing a message.

ML-KEM To establish a shared key between two users, ML-KEM takes advantage of three different algorithms. The first of these is a key generation algorithm that is probabilistic in nature. A key pair is generated, with one being kept private and the other being public. The second algorithm is the encapsulation algorithm, which is also probabilistic in nature. This algorithm results in the creation of a key and an associated ciphertext, and uses the public key created in the key generation algorithm. The ciphertext can then be sent to owners of the public keys used in the encapsulation algorithm. The owners of the public keys will have their corresponding private key, and can run the third function which is the decapsulation algorithm. This algorithm is deterministic in nature, and produces the shared key generated by the encapsulation algorithm while taking in the ciphertext and the private key as parameters.

2.3 Attribute-Based Encryption

An Access Control Mechanism (ACM), as defined by NIST[12], refers to the logical component that serves to receive the access request from the subject, to decide and then enforce the access decision. In this case, the subject can refer to any entity that desires to perform an action. It can be a user logging on to a system, an application trying to open a file, or a device connecting to a network. The Attribute-Based Access Control ACM receives the subject's access request and then evaluates the attributes of both subject and object against a specific policy. Here, the policy is a representation of relationships that aids in the evaluation of the subject's access request. When a user requests to access an object, the ACM evaluates these attributes and grants access based on the policy.

Traditionally, policy enforcement must rely on a trusted intermediary. Attribute-Based Encryption (ABE) removes this dependency by embedding policy enforcement within the cryptographic layer. There are two main variants of ABE that exist. Key-Policy ABE (KP-ABE) is a crypto-scheme in which ciphertexts are labeled with sets of attributes, and private keys are associated with access structures that control which ciphertexts a user is able to decrypt [11]. With

Ciphertext-Policy ABE (CP-ABE) [3], private keys are labeled with the sets of attributes while, ciphertexts are associated with access policies.

In [18], several industries are highlighted are shown to benefit from ABE. Privacy-critical applications are one of the major application areas of ABE because ABE allows fine-grained specification of user-centric access control to be embedded with the encryption. In healthcare, electronic health records can be encrypted with specific policies allowing only the relevant authorities to have access to them.

However, ABE has largely been ignored by the post-quantum community. A survey on post-quantum ABE highlights the need for more research on quantum resistant ABE to ensure the future viability of cryptographic protocols [13]. With the standardization of Covercrypt by ETSI, investigating ABE-based protocols is necessary to gain an understanding of their advantages in securing information systems.

2.4 Covercrypt

Covercrypt is a post-quantum public key encryption scheme that allows data to be encrypted for a set of attributes to provide fine-grained partitioning. It belongs to the family of CP-ABE schemes. To illustrate how Covercrypt works, we shall consider an enterprise that specializes in research and development. This organization has two departments: an AI department and the Robotics department. Additionally, this research company also has clearance levels that separate lower-level research material and employees from higher-level ones. Both departments and clearance levels are considered attributes that can be attached to specific users and documents. To be able to encrypt data for a set of attributes, Covercrypt needs a policy. Note that Cosmian and NIST do not share the same definition of a policy. A Covercrypt policy can be described as the existing space of rights used for encryption. In our use case, our policy would be represented by all four attributes, i.e., department AI and Robotics and clearance levels High and Low. This policy has two axes, the department axis, and the clearance level axis. Any combination of attributes can be defined as a partition. An access policy and an encryption policy are sets of these partitions. They can be represented as boolean expressions. The difference between the access policy and the encryption policy is that the encryption policy is stored with the ciphertext and determines who can access the ciphertext. The access policy is used to create a key, which is then used for decryption, which can occur only if the access policy satisfies the encryption policy.

Covercrypt provides the needed techniques to implement ABE, but it does not allow for attribute management and validation with relation to the users. Assigning attributes to valid users can be done when users are registered with the server, as stated in our assumptions earlier. However, an unauthenticated resource server has no guarantee that requests originate from a legitimate client, leaving it exposed to replayed queries from arbitrary parties. To this end, we add an authentication layer to the ABE techniques provided by Covercrypt to ensure that only verified clients can interact with the resource server. Each client

authenticates via a nonce-based challenge-response protocol signed with a client-held key, so the server only proceeds to key derivation and decryption once it has cryptographically confirmed it is speaking to a trusted client. Authorization at the user level is then enforced independently through Covercrypt’s attribute-based encryption: even an authenticated client can only decrypt data for which the requested identity’s attribute set satisfies the encryption policy, since the ABE key derivation itself is bound to that policy. This separates two concerns cleanly: the authentication layer establishes a trusted channel to the resource server, while the ABE layer enforces fine-grained, attribute-based access control over the data itself.

3 Overview of Protocol Design

The custom protocol discussed by the paper is a complete post-quantum protocol that provides security and privacy at the organizational level in an enterprise. All crypto-primitives used are post-quantum schemes standardized by either NIST or ETSI. Each crypto-primitive plays an important role in helping the protocol function as needed. The custom protocol features a post-quantum digital signature scheme that enables users to sign and verify messages. This helps in achieving mutual authentication. The next crypto-primitive used is ML-KEM, which allows for two parties to use a shared key without ever communicating the key over a network. The shared key generated using ML-KEM is used for symmetric encryption. Finally, Covercrypt provides attribute-based encryption for a set of attributes and serves as the main feature of this protocol. This protocol is designed to enable enterprises to encrypt and upload documents according to the internal structure of the organization. Documents uploaded can then be decrypted and downloaded if the user has the correct attributes.

The protocol can be divided into the upload stage and the download stage. The first stage of the protocol involves a user uploading a document to the enterprise server. This user, who will be referred to as Alice from now on, authenticates with the server by sending over her identity and nonce to it. The server responds with the signed nonces of Alice and the server. Alice then uses the server’s Covercrypt master public key to encrypt the document and its usage policy. The document is then stored by the server in cloud storage. The second stage of the protocol is the download stage. This stage involves another user, Bob, who wishes to download the documents that Alice has uploaded. Bob follows the same authentication pattern as Alice with the exchange of signed nonces. Using the ML-KEM public key of the server, Bob performs encapsulation and retrieves a ciphertext of a shared key for secret communication between Bob and the server. The ciphertext of the key $C_{B,S}$ is sent alongside the signed nonces (for Bob’s authentication) as well as the document’s id. The server is able to retrieve the attributes of Bob and evaluate the access policy that Alice specified in the Upload Phase to check Bob’s entitlement. The server then recovers the plaintext based on the user’s attributes. The following sections elaborate on the three major crypto-primitives used in the protocol and how they function

before summarizing the core concepts of implementing them in ProVerif and the protocol in Alice-Bob notation.

3.1 Threat Model and Assumptions

We consider an active adversary who can intercept, eavesdrop, and construct messages over the network. Cryptography is also assumed to be perfect and cannot be undone without the use of a key. For the protocol to work as expected, we assume that users have been pre-registered with the server (on-boarded). The security properties that we seek to verify are confidentiality, correctness of decryption, and mutual authentication. Through the use of queries, we are able to determine whether the protocol works as expected. Reachability queries like the one seen below allow us to check if certain states are reached by the model. An example of such a query can be found in the following snippet. The event `ServerAuthenticated` only fires when the signature of a message has been verified. The query then asks whether the event has ever been fired for a set of values.

Correspondence queries enable us to reason about relationships between two events in a trace. The query below requests for every instance of the event `PlaintextRetrieved(m)` occurring in a trace, a corresponding instance of the event `DocumentSent(m,ep)` having occurred earlier in the same trace. In plain terms, this means that every time Bob receives a document M and retrieves the plaintext, the server must have sent a document M to Bob beforehand in the same session. Thus, this guarantees authentication of the server to Bob.

4 Protocol Formalisation

In this section, we explain the protocol graphically and in Alice-Bob notation. This is an abstract presentation of the protocol that can be used as an intermediate step towards formalising it in ProVerif. Then, we present the ProVerif model explaining which security guarantees could be achieved and what that means.

4.1 Abstract Protocol in Alice-Bob Notation

A, B, S	: Alice, Bob, Server identities
K_S, K_A, K_B	: public signing keys (ML-DSA) of S, A, B
$K_S^{-1}, K_A^{-1}, K_B^{-1}$	: private signing keys (ML-DSA) of S, A, B
N_A, N_B, N_S, N'_S	: fresh nonces of A, B , and S (two sessions)
$K_{B,S}$	: symmetric key established via ML-KEM between B and S
$C_{B,S}$	: ML-KEM ciphertext sent from B to S
mpk	: Covercrypt master public key
P_M	: encryption policy attached to document M
M	: Some document stored at S to which Bob has access
id_M	: identifier for document M

We explain now in more depth how the authentication and privacy goals are realised by highlighting some of the ProVerif concepts used. Figure 1 summarizes the overall protocol in Alice-Bob notation. The protocol can be divided into two stages. The first stage is composed of Steps 1-3. Steps 4-7 represent the download stage.

1. $A \rightarrow S : (A, N_A)$
2. $S \rightarrow A : (\{N_A, N_S\}_{K_S^{-1}}, N_S)$
3. $A \rightarrow S : (\{N_A, N_S\}_{K_A^{-1}}, \{M, P_M\}_{mpk})$
4. $B \rightarrow S : (B, N_B)$
5. $S \rightarrow B : (\{N_B, N'_S\}_{K_S^{-1}}, N'_S)$
6. $B \rightarrow S : (\{N_B, N'_S\}_{K_B^{-1}}, C_{B,S}, id_M)$
7. $S \rightarrow B : (\{M\}_{K_{B,S}}, \{\{M\}_{K_{B,S}}, N_B, N'_S\}_{K_S^{-1}})$

Fig. 1. The protocol in Alice-Bob-notation

Upload phase While the Alice-Bob notation shows Alice and Bob communicating with a single server, the model utilizes two processes to represent a server instance that Alice and Bob communicate with. In Step 1, Alice begins the authentication process by sending her identity and nonce to the server. This marks the beginning of a new session and informs the server of the identity of the one starting the session. The second step involves the server responding by sending a signature over N_A, N_S using its ML-DSA private key. The server also sends its nonce in plaintext. Alice can verify the server's response using the server's ML-DSA public key. This step proves that the response came from the server and that the server received Alice's nonce. At the end of this step, the event **ServerAuthenticated** is triggered. This means that Alice knows that she is communicating with the server. Alice signs N_A, N_S using her private key and then sends it to the server who then verifies it using her public key. At the end of this step, if verification is successful, the **AliceAuthenticated** event is triggered and the server knows that it is communicating with Alice.

At this point in the protocol, both Alice and the server trust each other. Fresh nonces are generated for every session and signed together, so replay attacks are prevented because an attacker cannot forge a valid signature over a nonce pair from a future session. Any replayed signature is bound to stale nonces that won't match the current session's challenge. Alice encrypts her document with an encryption policy and the server's Covercrypt master public key, and sends the resulting ciphertext to the server to store. Alice also includes the nonces signed with her private key. After verification, the server stores the ciphertext, which triggers **DocumentUploaded**. As a symbolic verifier, ProVerif abstracts

away concrete data management. However, it provides us with tables, which represent its primary mechanism for modeling persistent state across protocol steps. Since the server is represented by two processes, the ciphertext is stored in the table by the process Alice communicates with, and then retrieved in the next phase by the process Bob communicates with.

Download Phase The next phase of the protocol involves the user Bob, who seeks to download the document that Alice uploaded. Bob repeats the authentication process Alice went through earlier; however, when sending the signature over N_B, N'_S to the server, Bob sends an ML-KEM-generated ciphertext and the id of the document he wants to retrieve. The ML-KEM ciphertext is generated by performing an encapsulation on the server's ML-KEM public key. This results in a ciphertext and a key that can be used for symmetric encryption. The equation enables ProVerif to reason that the encapsulation operation and decapsulation operation produce the same symmetric key. User attributes are linked to the user's identity, which enables the resource server to evaluate the access policy of Bob. This access policy is used to create a unique Covercrypt decryption key. This decryption key is used to decrypt the ciphertext uploaded by Alice. When decryption succeeds, an event called `DecryptionSucceeded(ap,ep)` is fired. This allows us to track which encryption policy and access policy worked. To ensure ProVerif reasons correctly about decryption, each valid pair of encryption and access policy has their own destructor. The recovered plaintext is re-encrypted by using the shared key generated by the decapsulation operation on the server side, and then sent to Bob. Bob is able to recover the original plaintext and now has access to the document. This triggers the `PlaintextRetrieved` event in the model.

4.2 Verification Results

In this section, we analyse and interpret the results of the ProVerif formalisation and verification.

Mutual Authentication One of the desired properties of this protocol is mutual authentication. During the upload phase, Alice and the server prove their identity to each other. Likewise, in the download phase, Bob and the server also prove their identity to each other. To verify mutual authentication in ProVerif, we can take advantage of injective correspondence queries. In addition to ensuring earlier appearances of an event in a trace, these queries guarantee a unique earlier appearance of the second event. These queries ensure that for every instance of `AliceAuthenticated(na,ns)` (as well as `BobAuthenticated(na,ns)`) there is *only one* corresponding instance of `ServerAuthenticated(na,ns)` (for Alice and Bob, respectively) earlier in the trace. This means that the protocol guarantees authentication of Alice and Bob to the server and, because of the uniqueness of the previous message, it is also not susceptible to replay attacks.

```
Query inj-event(AliceAuthenticated(na,ns)) ==>
      inj-event(ServerAuthenticated(na,ns)) is true.
```

Confidentiality We need to be sure that the attacker cannot have access to any secret document. To verify confidentiality, only a single query was needed. There are two types of attackers in ProVerif. Passive attackers only observe; the attacker model is mainly used when checking for secrecy. Active attackers can send and intercept messages. The query below checks if the attacker can construct the secret document based on the information it has observed from the communication between Alice, the server, and Bob. The result proves that the secret document is kept safe from the attacker since they are unable to reconstruct the secret document based on observable knowledge.

```
Query not attacker(secret_doc[]) is true.
```

Correctness of Decryption The next security property to verify is the correctness of decryption. Since the protocol uses an ABE scheme, decryption must work as the logic of the scheme dictates. To ensure correctness, we track a specific event that occurs within the model. DecryptionSucceeded only fires when a successful decryption occurs. What makes this event useful is the parameters it takes. By taking both access policies and encryption policies as parameters, it makes it easier to query which pairs would produce a successful event. An example of a matching pair can be found in the following snippet. The encryption policy and access policy match; therefore, decryption occurs and the event fires.

```
Query not event(DecryptionSucceeded(and_policy(dept_ai,high),
                                             and_enc_policy(dept_ai,high))) is
      false.
```

Another scenario to consider is when decryption does not occur due to the difference in encryption and access policies below. The access policy has a clearance level of low, while the encryption level requires a clearance level of high. The event does not fire; hence the query evaluates to true.

```
Query not event(DecryptionSucceeded(and_policy(dept_robotics,low),
                                             and_enc_policy(dept_robotics,high))) is true.
```

To verify the correctness of decryption when the encryption policy contains two departments and a clearance level, the following type of query is used. This is meant to model how Covercrypt uses the logical OR when there are two or more non-hierarchical attributes in the same policy axes.

```
Query not event(DecryptionSucceeded(and_policy(dept_ai,high),
                                             compound_enc_policy(and_enc_policy(dept_ai,high),
                                             and_enc_policy(dept_robotics,high)))) is false.
```

5 Conclusions

As companies begin planning their migration towards post-quantum security, preparations need to be made to protect their systems in the long term. This paper presents a post-quantum protocol by combining several post-quantum schemes to achieve post-quantum security in each step of the protocol. The cloud server must be completely trusted for the protocol to work as intended. Apart from being one of the few protocols which provide both post-quantum and attribute-based encryption, there are other benefits to this protocol. Data is quantum-safe before it is stored in the database, making it impossible for any attacker to gain anything meaningful from compromising the database. The usefulness of Covercrypt depends on the authorization and the authentication system to which it is linked. If an attacker is able to impersonate a valid user, the security guarantees it provides no longer matter. Covercrypt can only be integrated within a protocol with a strong and secure authentication system.

Despite the guarantees formal verification gives regarding the protocol, there are some weaknesses within the protocol. Most communication protocols have fallback modes that are used when issues occur. This protocol lacks that, meaning when problems arise, either due to unforeseen examples, the entire protocol would fail. It will be prudent to conduct more research on properties that pertain to resilience in future work. The custom protocol presented in this paper focuses on providing a post-quantum secure protocol. Performance issues are not a primary concern, although following the principles of good protocol design, we minimise the use of computationally expensive cryptographic algorithms. However, further refinements of the abstract protocol presented here are needed to add additional performance optimizations. For example, in Step 7 in Figure 1 the entire ML-KEM encrypted message together with the nonces, is signed, while realistically we would only sign a hash of these items.

Implementing the custom protocol can be seen as the next step to take. Covercrypt has released a library in multiple languages that allows developers to take advantage of post-quantum ABE. This library was used in an initial feasibility test [10] in which conventional TLS was used instead of the ML-KEM and ML-DSA combination. The former iteration of the protocol was built using Java Spring Boot and React. Additionally, Amazon S3 and PostgreSQL database were used to store the encrypted blobs and important metadata, respectively. A Render server was used to host both frontend and backend applications. This version of the custom protocol used Covercrypt’s Java library. To implement this protocol in its current iteration would require the use of OpenQuantumSafe’s ML-KEM project and IBM’s ML-DSA implementation.

Although ProVerif was useful in modeling certain aspects of the protocol, there are some downsides that made it a challenge. Firstly, some attack traces do not directly map to a real-world attack. This led to extra precautions being taken to satisfy the model checker. Also, there was the issue of state management, which was a bit difficult to model. Using tables served as a useful intermediary; however, their use in the model caused it to deviate a bit from how state management would work in the real world. Despite these challenges, the model checker

was still useful in verifying desirable properties. Using ProVerif, we establish the confidentiality of documents at rest, correctness of attribute-based decryption, and mutual authentication.

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

References

1. Ajtai, M.: The shortest vector problem in l_2 is NP-hard for randomized reductions. In: STOC, pp. 10–19. ACM (1998). <https://doi.org/10.1145/276698.276705>
2. Baier, C., Katoen, J.P.: Principles of Model Checking. MIT Press (2008)
3. Bethencourt, J., Sahai, A., Waters, B.: Ciphertext-policy attribute-based encryption. In: IEEE Symposium on Security and Privacy, pp. 321–334 (2007). <https://doi.org/10.1109/SP.2007.11>
4. Brézot, T., de Perthuis, P., Pointcheval, D.: Covercrypt: An efficient early-abort KEM. In: ESORICS 2023, pp. 372–392. Springer (2024). https://doi.org/10.1007/978-3-031-50594-2_19
5. Bhargavan, K., Jacomme, C., Kiefer, F., Schmidt, R.: Formal verification of the PQXDH post-quantum key agreement protocol. In: USENIX Security 2024, pp. 469–486 (2024)
6. Blanchet, B.: Automatic verification of security protocols in the symbolic model: the verifier ProVerif. In: FOSAD 2014. LNCS, vol. 8604, pp. 54–87. Springer (2014)
7. Dolev, D., Yao, A.C.: On the security of public key protocols. In: FOCS, pp. 350–357 (1981). <https://doi.org/10.1109/SFCS.1981.32>
8. Evans, R.W.N.O.: Designing and formally verifying a post-quantum protocol. Master’s thesis, Middlesex University London (2026)
9. Evans, R.W.N.O.: Protocol model (2026). https://github.com/NiiOdartei7/quantum_model
10. Evans, R.W.N.O.: QuantumSpringBoot (2026). <https://github.com/NiiOdartei7/QuantumSpringBoot>
11. Goyal, V., Pandey, O., Sahai, A., Waters, B.: Attribute-based encryption for fine-grained access control. In: CCS, pp. 89–98 (2006). <https://doi.org/10.1145/1180405.1180418>
12. Hu, V.C., Ferraiolo, D., Kuhn, R., et al.: Guide to attribute based access control. NIST SP 800-162 (2014). <https://doi.org/10.6028/NIST.SP.800-162>
13. Jemihin, Z.B., Tan, S.F., Chung, G.C.: Attribute-based encryption in securing big data from post-quantum perspective. Cryptography **6**(3), 40 (2022). <https://doi.org/10.3390/cryptography6030040>
14. Micciancio, D., Regev, O.: Lattice-based cryptography. In: Post-Quantum Cryptography, pp. 147–191. Springer (2009). https://doi.org/10.1007/978-3-540-88702-7_5
15. National Institute of Standards and Technology: Module-lattice-based digital signature standard. FIPS 204 (2024). <https://doi.org/10.6028/NIST.FIPS.204>
16. Prévost, T., Martin, B., Alibert, O.: ETSI GS QKD compliant TLS implementation. In: SECURE, pp. 705–710 (2025). <https://doi.org/10.5220/0013564700003979>
17. Regev, O.: On lattices, learning with errors, random linear codes, and cryptography. In: STOC, pp. 84–93. ACM (2005). <https://doi.org/10.1145/1060590.1060603>
18. Zickau, S., et al.: Applied attribute-based encryption schemes. In: ICIN (2016)

Kleptographic SETUP against Pseudonym Unlinkability or: check your EDIW Wallet

Przemysław Błażkiewicz¹[0000–0001–5570–5938], Anna
Lauks-Dutka¹[0000–0002–9165–1110], Mirosław Kutylowski²[0000–0003–3192–2430],
and Przemysław Kubiak²[0000–0002–1349–093X]

¹ Wrocław University of Science and Technology

{przemyslaw.blaskiewicz, anna.lauks}@pwr.edu.pl

² National Research Institute, Poland

{mirosław.kutylowski, przemysław.kubiak}@nask.pl

Abstract. The eIDAS Regulation guarantees users of the European Digital Identity Wallet (EDIW) the right to use unlinkable pseudonyms generated by the wallet. The EU Implementing Act CIR 2025/2979 points to WebAuthn as a standard for creation and use of pseudonyms on EDIW.

We show that a corrupted EDIW wallet may create a kleptographic proof that links two different pseudonyms. Such a wallet remains cryptographically indistinguishable from an honest one, yet a third party holding a kleptographic secret key can extract and verify a proof hidden inside a regular WebAuthn signature created for proving ownership of those pseudonyms.

This proof is an undeniable evidence that the secret keys corresponding to two WebAuthn pseudonyms reside on the same wallet. The kleptographic setup works for widely used signature schemes based on the discrete logarithm problem.

The problem affects both lightweight implementations by a *wallet instance* (with limited subversion protection by technical means) and those based on secure elements (like WSCD), where the setup can be hidden, say, for the sake of “national security”.

Keywords: pseudonym · WebAuth · European Digital Identity Wallet · unlinkability · kleptography · ECDSA · EC Schnorr

1 Introduction

Pseudonymization is one of the fundamental tools for personal data protection under the EU General Data Protection Regulation [1]. It is the first step to converting *personal* to *non-personal data*. Moreover, under the data minimization principle set out in the GDPR, explicit identifiers should be used only when necessary for data processing.

Pseudonymization should not be understood merely as using anonymous identifiers: we may need a digital proof that the pseudonym belongs to the person using it, i.e., pseudonymous authentication and signature schemes. In this scenario, a separate pseudonymous key pair (pk_D, sk_D) for a user is dedicated to each context/application case D referred to as a *domain*. Then, pk_D can serve as the pseudonymous identifier (equivalently, a handle to pk_D can be treated as the pseudonym).

1.1 Unlinkability

The key property for privacy protection via pseudonymization is *unlinkability*. Its exact meaning is hard to pin down and the situation resembles that of defining *privacy*. On the one hand, the unlinkability of pseudonyms pk_{D_1} and pk_{D_2} for different domains D_1, D_2 should mean that the relying parties cannot determine whether pk_{D_1} and pk_{D_2} belong to the same person based on interactions authenticated with pk_{D_1} and pk_{D_2} . However, this claim is limited to what can be drawn from executions of the cryptographic procedures, and does not take into account side information (e.g., timing data).

In the case of a single domain D , unlinkability is no longer a requirement having the same meaning for all application areas. In some cases (cf. Privacy Pass), a user should use multiple unlinkable pseudonyms within the same domain. However, for applications such as providing pseudonyms for whistleblowers, we need to prevent Sybil attacks and a user must have at most one pseudonym per domain. The situation becomes complicated when the pseudonym is generated or certified by an issuer such as a certificate authority. In this case, unlinkability may mean that the issuer cannot link a pseudonym (confirmed by a certificate issued by this authority) to a physical person. This requires blind issuance of certificates. On the other hand, the issuance process has to guard against misuse by the users (e.g., a minor obtaining a certificate for an adult person). At the same time, perfect unlinkability creates security problems: it prevents prosecuting a pseudonymous user committing a crime by hiding the link to the real identity. It even prevents a malicious user from being blacklisted across domains.

1.2 Pseudonyms within the eIDAS framework

The original EU eIDAS Regulation No. 910/2014 [2] established the European framework for electronic identification and trust services. This framework was substantially amended by EU Regulation 2024/1183[3], which establishes the European Digital Identity Framework and introduces the European Digital Identity Wallet (EDIW). The amended eIDAS Regulation explicitly addresses the use of pseudonyms: Article 5 states that, without prejudice to Union or national law requiring identification of users, the use of user-chosen pseudonyms shall not be prohibited. Consequently, pseudonyms remain a legally recognised means of accessing electronic services whenever identification by legal identity is not required. The pseudonyms may appear in qualified certificates for advanced electronic signatures, in qualified electronic attestations of attributes, and in electronic attestations of attributes issued by or on behalf of public bodies responsible for authentic sources.

The amended eIDAS Regulation introduces European Digital Identity Wallets as a central component of the European Digital Identity Framework. According to Article 5a, each Member State shall provide at least one EDIW, enabling users to obtain, store, manage, and present person identification data (PID) and electronic attestations of attributes (EAA) under their sole control. According to Recital 22, EDIW “*should include a functionality to generate user-chosen and managed pseudonyms, to authenticate when accessing online services.*” Article 5a.4, states that EDIW “*shall enable the user, in a manner that is user-friendly, transparent, and traceable by the user, to: ... (b)*

generate pseudonyms and store them encrypted and locally within the European Digital Identity Wallet; ”.

Regarding unlinkability, Article 5 a³: “16. *The technical framework of the European Digital Identity Wallet shall:*

(a) not allow providers [...] to obtain data that allows transactions or user behaviour to be tracked, linked or correlated, or knowledge of transactions or user behaviour to be otherwise obtained, unless explicitly authorised by the user;

(b) enable privacy preserving techniques which ensure unlikeability, where the attestation of attributes does not require the identification of the user.”

The phrase “explicitly authorised by the user” should be understood as permitting the use of the same or linkable user identifiers – pseudonymous or not. In turn, point (b) underlines that PET technologies should be available for the user. No further details are given in the regulation, no particular pseudonymity model is defined at this point.

The legal requirements are complemented by the European Digital Identity Architecture and Reference Framework (ARF) [4], which defines high-level requirements for the EDIW ecosystem and serves as a foundation for the technical specifications, standards, and procedures referenced by the European Commission in the implementing acts of the European Digital Identity Regulation. Some information on the pseudonymization concept for EDIW can be found in the implementing Regulation [5]. According to Article 14, pseudonyms can be used by wallet users and by relying parties. Moreover, users’ pseudonyms should comply with Annex V, which points to “WebAuthn – W3C Recommendation, 8 April 2021, Level 2” (see [6]). Cryptographically, WebAuthn is based on domain-specific key pairs generated independently by the authenticator (in our case, the wallet). These keys are used in a challenge-response protocol, where a random nonce chosen by the server is signed by the authenticator. The public key and the IDs generated during the registration process serve as pseudonyms (see Sect.14.2 and Sect.14.6.2 in [6]).

One of the key technical questions is how to generate, store, and use the private key used by WebAuthn. While certain assets of an EDIW are protected by the Wallet Secure Cryptographic Device (WSCD) and the Wallet Secure Cryptographic Application (WSCA), the obligation to use them is confined to “critical assets”. The legal definition of critical assets ([5], Article 2) reads as follows: “‘critical assets’ means assets within or in relation to a wallet unit of such extraordinary importance that where their availability, confidentiality or integrity are compromised, this would have a very serious, debilitating effect on the ability to rely on the wallet unit”. Whether pseudonyms’ unlinkability is a critical asset may depend heavily on the application area. For critical scenarios, such as filing whistleblower reports or reporting organized crime, strong unlinkability is essential. In this case, the private keys should be managed and protected by high-security WSCA and WSCD components. However, for low-risk activities within a local discussion forum or a fan club, moving the implementation of pseudonyms into WSCA and WSCD would overshoot the target. Consequently, at least some pseudonyms might be implemented in the less secure part of the EDIW wallet such as a smartphone-based *wallet instance*.

³ English official text contains the word “unlikeability” instead of “unlinkability”. The Polish and German versions refer to unlinkability (*uniemożliwienie powiązania*, *Unverknüpfbarkeit*).

Clearly, the landscape of pseudonyms is far from well-established, and there is a vivid ongoing discussion on pseudonymization within the ARF initiative (for details, see the GitHub repository for the ARF [7]). In what follows, we present another critical issue that needs to be addressed.

1.3 Kleptography and Anamorphism

In this paper, we focus on covert functionalities that may be hidden in a cryptographic device allegedly implementing a legitimate protocol. The resulting modified protocol is usually called *kleptographic*, if the target is to leak private information from the device to the adversary holding specific private kleptographic key, or *anamorphic* if it enables bypassing censorship. The following properties are fundamental for the design of such schemes:

Indistinguishability: No computationally bounded distinguisher can differentiate with a non-negligible advantage between a device executing protocol P and its modified version P' . This holds even if the distinguisher knows the code of P' and its public kleptographic keys (in the case of kleptography) and the private keys officially possessed by the device (in the case of anamorphism).

Restricted access to hidden functionality: Only the adversary Adv holding a specific private key (called kleptographic private key or dual key in the case of anamorphic schemes) can take advantage of covert functionalities provided by P' . In particular, the private kleptographic key is needed to access the data leaked by P' .

Therefore, unless a device implementing P' is reverse-engineered, the attack remains undetectable to the user. And even if reverse engineering exposes additional functionalities, it does not help detect modifications on other devices allegedly implementing legitimate protocol P .

In the classical setting, kleptography and anamorphism focus on creating a hidden communication channel. We go beyond that and consider the following two scenarios:

passive scenario: the modified protocol P' offers only an additional data communication channel,

active scenario: the modified protocol P' in fact executes an additional cryptographic protocol. In our case, it will be a proof of possession of two secret keys, rather than merely a declaration about it.

Kleptography/anamorphism versus pseudonym unlinkability. When talking about hidden functionalities concerning pseudonyms on EDIW, the fundamental question is about their purpose and beneficiary. While the word *kleptography* has immediate negative connotations (*stealing, theft,...*), it may serve good purposes. In the context of pseudonyms, we may consider the following legitimate uses of kleptography that involve breaking unlinkability:

Pseudonymous account ownership: A user may lose the pseudonymous private key (e.g., due to a technical failure). In such cases, an undeniable proof linking the pseudonym to its owner, or to another owner's pseudonyms, might be crucial for recovering access to an account.

Blacklisting: It might be necessary to blacklist a malicious user or all pseudonyms managed by a stolen device. This requires a reliable proof of pseudonym possession to be obtained by a trusted authority.

Law enforcement: Pseudonyms can be used by people engaged in illicit activities (e.g., child abuse). In this case, breaking unlinkability with kleptographic methods would enable the identification of offenders and discourage criminal activities.

Note that the European Commission is making plans to facilitate law enforcement by breaking cryptographic protection [8]. The EU Roadmap explicitly mentions decryption, but deanonymization appears to be equally important in achieving these targets.

On the other hand, hidden functionalities might be misused for malicious purposes:

Illusory privacy protection: a provider of devices implementing pseudonyms can install a modified protocol P' enabling linking the pseudonyms originating from the same device and/or linking them with the device owner. Then a user would have no idea that privacy guarantees have been violated and, in fact, would be even more exposed to privacy threats than without pseudonyms. If the additional functionality provides undeniable proof for third parties, then the links between pseudonyms may be offered for sale on a black market or made available to enemy countries.

As we see, the situation is very delicate, and great care is needed when admitting devices that implement pseudonyms. In this paper, we show the technical feasibility of kleptographic schemes against unlinkability. So, the discussion above concerns a technically feasible threat rather than merely a hypothetical scenario.

1.4 Previous Work

Several kleptographic attacks have been considered for WebAuthn used by FIDO2 in [9]. Among others, they enable linking different public keys as keys originating from the same authenticator. However, because a compromised authenticator can falsify the owner's identity, these specific leaked proofs cannot be trusted for tasks such as account recovery. It should be noted that the threat to FIDO2 is limited as long as it is implemented on honest and tamper-evident hardware tokens.

Restricted Identification, the precursor of pseudonymous identification systems, was implemented in German personal identity cards in 2010 [10]. It is limited to creating unlinkable pseudonyms without signatures, relying entirely on the card's security mechanisms. Interestingly, [10] introduces two pseudonym variants derived as $\text{Hash}(D^{\text{sk}})$, where sk is the card's secret key devoted to this purpose and D is a domain-specific group element. D can either be a pseudorandom element with unknown discrete logarithm, which makes linkability infeasible, or its logarithm may be known collectively by a group of authorities, allowing them to check if a pseudonym belongs to a specific person. The use of Hash prevents the identity associated with a given pseudonym to be found differently than by trying all candidates one by one. The ID card itself cannot prove that two pseudonyms originate from the same device, unless communication is physically monitored in an isolated environment.

The signature-in-signature concept used in this paper was discussed in [11] and [12]. The purpose was different: fishing out forged signatures in case of a key compromise.

2 Kleptographic proof of possession of two secret keys

We demonstrate that the WebAuthn signatures can carry linking information in a kleptographic way proving that WebAuthn alone is insufficient to guarantee the unlinkability features required by eIDAS. Consequently, the certification of EDIW should take this issue into account – both to prevent the malicious use and, potentially, to enable controlled leakage of linking information for purposes mentioned in Section 1.3. The construction below serves as a starting point rather than an ultimate solution. Our aim is to initiate a discussion about unlinkability within the EDIW ecosystem, given the adoption of WebAuthn in [5]. For simplicity, we focus on ECDSA. However, designing kleptographic linking for post-quantum pseudonyms might be the next important step.

Assumptions. We briefly outline below our assumptions regarding the ecosystem of pseudonyms:

Secure key generation: We assume that the key generation process is free from kleptographic subversion. First, generating a pseudonymous key pair is a one-time event that can be well-protected (e.g., via watchdogs or distributed key generation). Second, eIDAS states that pseudonyms can be freely chosen by the user.

Standard schemes: Kleptographic schemes should be designed for widely used standard schemes.

Attack target: The kleptographic leakage should be a proof that a given device knows secret keys sk_1, sk_2 corresponding to public keys (pseudonyms) PK_1, PK_2 . This is the ultimate proof that these pseudonyms belong to the same person.

2.1 Schnorr/ECDSA Signature

EC-Schnorr signatures. For the reader’s convenience, we briefly recall the Schnorr signature scheme on elliptic curves as it is defined in [13] (simplifying some less relevant details). The scheme uses a set of carefully chosen parameters: an elliptic curve E and a point $G \in E$ of prime order q . We assume that the private signing key sk has been chosen uniformly at random and that $PK = sk \cdot G$ is the public verification key. A signature on M is calculated as described in Algorithm 1.

Algorithm 1 EC-Schnorr Signature Creation with a Random Nonce

Require: signing private key sk , message M , random nonce k

procedure EC – Schnorr(sk, M, k)

$R \leftarrow k \cdot G$

$e \leftarrow \text{Hash}(R_x, R_y, M)$ where R_x, R_y denote, respectively, the x and y -coordinate of R

$s \leftarrow k + sk \cdot e \bmod q$

return signature (s, e)

ECDSA signatures. In Algorithm 2, we recall the ECDSA signature according to [13] with some details omitted. The parameters G, q play the same role as for EC-Schnorr.

Algorithm 2 ECDSA Signature Creation with a Random Nonce

Require: signing private key sk , message M , random nonce k

procedure ECDSA(sk, M, k)

$R \leftarrow k \cdot G$

$k_{\text{inv}} \leftarrow k^{-1} \bmod q$

$s \leftarrow k_{\text{inv}} \cdot (R_x \cdot \text{sk} + \text{Hash}(M)) \bmod q$ where R_x denotes the x -coordinate of R

return signature (R_x, s)

Knowledge of ephemerals. Security of Schnorr signatures was considered in the Random Oracle Model (ROM) using the Forking Lemma [14]: the ability to forge signatures in ROM implies the ability to derive the signing key sk . Analogous argument holds for ECDSA. It follows that in practice, we can make the following assumption:

Assumption 1 *If a device dev can create a valid EC-Schnorr signature (s, e) verifiable with the public key pk , then dev can derive the ephemeral k such that $R = k \cdot G$ was used during signature creation.*

Analogously, if D creates an ECDSA signature (R_x, s) , then dev can derive the ephemeral k such that $R = k \cdot G$.

Indeed, as dev must know sk , it can derive k from the equation $s = k + \text{sk} \cdot e \bmod q$ in the case of the EC-Schnorr signature. For ECDSA, it is also very easy: k_{inv} can be calculated from $s = k_{\text{inv}} \cdot (R_x \cdot \text{sk} + \text{Hash}(M)) \bmod q$, and then inverted modulo q .

2.2 Kleptographic Signature in Signature

Our construction requires a secret key Dkey shared by the device dev and adversary Adv . For example, it might be derived using static Diffie-Hellman key exchange with the share $Z = z \cdot G$ of dev (sent during an arbitrary DH key exchange observable by Adv) and the public kleptographic key PK_{Adv} of the adversary: $\text{Dkey} \leftarrow z \cdot \text{PK}_{\text{Adv}}$, and equivalently, $\text{Dkey} \leftarrow \text{sk}_{\text{Adv}} \cdot Z$.

For each signature creation process, dev will need a dedicated ephemeral key $U = u \cdot G$, where u is known to the device dev and U is known to Adv . For example, U might be the DH key exchange share generated by dev in the last session. The necessary condition is that reusing u does not compromise the security of other procedures. This holds for an ephemeral share of the DH key, as the key exchange can be simulated using U after u is chosen for the target protocol.

Kleptographic signature creation. Algorithm 3 embeds an EC-Schnorr signature (created using sk_2) into the ephemeral k of the ECDSA signature (created using sk_1). As long as this component k can be treated as a pseudorandom one, the resulting ECDSA signature should preserve its properties. In turn, k can be treated as pseudorandom if the element e can be treated as random. This is possible, as e is a hash value computed from arguments involving pseudorandom secret S shared only by device dev and adversary Adv and unique for each signature creation process. The counter ctr should be stored in non-volatile memory of dev .

Algorithm 3 ECDSA Kleptographic Signature Creation

Require: signing private keys sk_1, sk_2 , message M , random ephemeral u with $U = u \cdot G$ known to Adv, shared key Dkey, internal counter ctr

procedure lnk-ECDSA($\text{sk}_1, \text{sk}_2, M, u, \text{Dkey}, \text{ctr}, U$)

▷ Start of Phase 1

increment ctr

$S \leftarrow \text{Hash}(\text{Dkey}, \text{ctr})$

$e \leftarrow \text{Hash}(U_x, U_y, S, M)$ where U_x, U_y denote the x and y -coordinate of U

$k \leftarrow u + \text{sk}_2 \cdot e \bmod q$

▷ Start of Phase 2

$R \leftarrow k \cdot G$

$k_{\text{inv}} \leftarrow k^{-1} \bmod q$

$s \leftarrow k_{\text{inv}} \cdot (R_x \cdot \text{sk}_1 + \text{Hash}(M)) \bmod q$ where R_x denotes the x -coordinate of R

return signature (R_x, s)

Construction rationale. Let us note that a device dev that creates a signature with the structure described in Algorithm 3 requires both sk_1 and sk_2 , corresponding to the public keys PK_1, PK_2 , respectively, which serve as two different pseudonyms. Later, we provide a more formal argument, but the idea is as follows: since dev creates an ECDSA signature verifiable using PK_1 , it must know sk_1 . In turn, by Assumption 1, it knows the ephemeral k . Furthermore, e can be computed from values shared by dev and Adv, and u is known to dev. Consequently, the device dev can recalculate sk_2 from the equation $k = u + \text{sk}_2 \cdot e \bmod q$.

Kleptographic signature verification. The verification of the kleptographic signature by Adv is presented in Algorithm 4. The first part is the regular ECDSA verification. If this signature is valid, then Adv can conclude that dev has access to sk_1 . Moreover, according to Assumption 1, Adv can assume that dev knows the ephemeral k such that $R = k \cdot G$. So, Adv has to check that k is a valid second component for an EC-Schnorr signature created with sk_2 without seeing k . Instead of checking the properties of k (as it is done for EC Schnorr), Algorithm 4 verifies the properties of $R = k \cdot G$ already recalculated as $R \leftarrow u_1 \cdot G + u_2 \cdot \text{PK}_1$. Adv has to check that $k = u + \text{sk}_2 \cdot e \bmod q$. Since the group used has the prime order q , this is equivalent to check that $k \cdot G = u \cdot G + \text{sk}_2 \cdot e \cdot G$, which translates to $R \stackrel{?}{=} U + e \cdot \text{PK}_2$.

Algorithm 4 is performed by Adv locally, leaving only him convinced of the verification result. However, using standard techniques, this verification process can be converted into a non-interactive zero-knowledge proof that can be presented to a third party.

3 Draft of Security proofs

3.1 Indistinguishability

We consider a distinguisher holding $\text{PK}_1, \text{PK}_2, U$ and an ECDSA signature (R_x, s) , whose goal is to determine how the signature was created. In the underlying security game, the challenger randomly chooses how to create a signature:

Algorithm 4 ECDSA Kleptographic Signature Verification

Require: public keys PK_1, PK_2 (pseudonyms), message M , ECDSA signature (r, s) , random point U , shared key $Dkey$, an estimated interval T containing the current value of ctr

procedure `ver-link-ECDSA`($PK_1, PK_2, M, (r, s), Dkey, U, T$)
 $\triangleright$ *Start of verification of (r, s) with PK_1 and M*
 $s_{inv} \leftarrow s^{-1} \bmod q$
 $u_1 \leftarrow s_{inv} \cdot \text{Hash}(M) \bmod q$
 $u_2 \leftarrow s_{inv} \cdot r \bmod q$
 $R \leftarrow u_1 \cdot G + u_2 \cdot PK_1$
 accept (r, s) if $R_x = r$ where R_x denotes the x -coordinate of R
 $\triangleright$ *Start checking the proof of possession of sk_2*
 Find $ctr \in T$ such that for:
 $S \leftarrow \text{Hash}(Dkey, ctr)$
 $e \leftarrow \text{Hash}(U_x, U_y, S, M)$ where U_x, U_y denote the x and y -coordinate of U
 the condition $R = U + e \cdot PK_2$ is satisfied
 if such ctr is found then **return** linked else **return** not-linked

regular execution: according to the original specification of ECDSA,

kleptographic execution: according to Algorithm 3.

A distinguisher $\mathcal{D}$ holding PK_1, PK_2, U and the resulting signature has to answer which option was chosen by the challenger. For simplicity, we consider only a single signature; generalization to an arbitrary number of signatures is straightforward. We have to show that the advantage of $\mathcal{D}$ is negligible. For this purpose, we construct a sequence of games.

Game 0: it is the protocol from Algorithm 3.

Game 1: $Dkey$ is replaced by a random element V . If there is a non-negligible difference between Game 0 and Game 1, then it would be possible to construct a distinguisher that solves the DDH Problem with a non-negligible advantage. In that case, the DDH input would be used as (G, Z, PK_{Adv}, V) and the protocol execution would be simulated for sk_1, sk_2, u chosen by the simulator.

procedure GAME 1

$V \leftarrow_{\mathcal{S}} \mathcal{G}$
 $S \leftarrow \text{Hash}(V, t)$
 $e \leftarrow \text{Hash}(U_x, U_y, S, M)$
 $k \leftarrow u + sk_2 \cdot e \bmod q$
 $R \leftarrow k \cdot G$
 $k_{inv} \leftarrow k^{-1} \bmod q$
 $s \leftarrow k_{inv} \cdot (R_x \cdot sk_1 + \text{Hash}(M)) \bmod q$
return signature (R_x, s)

Game 2: The assignment $S \leftarrow \text{Hash}(V, ctr)$ is replaced by a random choice of S . The games 2 and 1 are not distinguishable as long as Hash is correlated-input secure hash function [15].

Game 3: As S is chosen at random in Game 2, we can replace e with a random element, again assuming that Hash is correlated-input secure hash with image $\mathbb{Z}_q$. Then the game can be reduced to the following form, and the advantage of $\mathcal{D}$ will change by a negligible value:

```

procedure GAME 3
   $e \leftarrow_{\S} \text{Image}(\text{Hash})$ 
   $k \leftarrow u + \text{sk}_2 \cdot e \bmod q$ 
   $R \leftarrow k \cdot G$ 
   $k_{\text{inv}} \leftarrow k^{-1} \bmod q$ 
   $s \leftarrow k_{\text{inv}} \cdot (R_x \cdot \text{sk}_1 + \text{Hash}(M)) \bmod q$ 
  return signature  $(R_x, s)$ 

```

Game 4: As e is random, we can calculate k right away as random. The advantage of the distinguisher does not change. However, after the change, we obtain the regular ECDSA signing process for the key sk_1 .

3.2 Unforgeability of kleptographic signature

It suffices to show that an adversary Adv' holding sk_{Adv} and at most one of the keys sk_1, sk_2 cannot create a signature that passes the verification process from Algorithm 4. This covers the case of kleptographic adversary Adv , who, in the worst case, is a user and holds one of the keys (thus, we assume that sk_1 and sk_2 do not originate from the same device).

First, assume that Adv' knows only sk_2 and sk_{Adv} , but can nevertheless create a kleptographic signature successfully verified by Algorithm 4. In that case, Adv' would have a peculiar method of forging signatures verifiable with PK_1 . Thus, ECDSA would be broken. As we assume ECDSA to be secure, Adv' must fail in this case.

Now assume that Adv' knows $\text{sk}_1, \text{sk}_{\text{Adv}}$ and can forge a kleptographic signature (R_x, s) successfully verified by Algorithm 4. The verification of ECDSA ensures that $s = k_{\text{inv}} \cdot (R_x \cdot \text{sk}_1 + \text{Hash}(M)) \bmod q$. So, using sk_1 , the adversary Adv' can calculate k_{inv} , and therefore k . Since $R = U + e \cdot \text{PK}_2$ (the condition checked by Algorithm 4), we must have $k = u + \text{sk}_2 \cdot e \bmod q$. On the other hand, Adv' can calculate S , and therefore $e \leftarrow \text{Hash}(U_x, U_y, S, M)$. Thus, Adv' would be able to extract a valid EC-Schnorr signature (k, e) for message (S, M) , thereby forging a EC-Schnorr signature for PK_2 . As we assume EC-Schnorr to be secure, Adv' must fail also in this case.

3.3 Security against Adv

The argument from Section 3.2 does not exclude the situation in which Adv after obtaining kleptographic ECDSA signatures, launches an attack different from generating yet another signature valid in the sense of Algorithm 4. Consider such an attack $\mathcal{A}$. The point is that we can mimic the attack for an adversary Adv' that gets the ECDSA signatures created in the regular way. Let (R_x, s) be an ECDSA signature created in

this way. By reprogramming Random Oracle for Hash, we can convert it into a kleptographic ECDSA signature using an approach similar to security arguments for Schnorr signatures:

1. reconstruct point R such that its x -coordinate is R_x (as it is done during the regular ECDSA verification process),
2. choose e at random from $\mathbb{Z}_q$, the image of Hash,
3. calculate $U \leftarrow R - e \cdot PK_2$,
4. insert e as the hash value $\text{Hash}(U_x, U_y, S, M)$ in the Random Oracle table for Hash.

Note that the second step is possible, because Algorithm 3 assures that the hash argument S is fresh and Hash is correlated-input secure. Moreover, U will not repeat.

In this way, we actually do not derive u , but we assume that U can be used for DH key exchange only, where the session key can be derived by emulating the other protocol participant.

After the conversion(s), the attack $\mathcal{A}$ can be applied by Adv, who emulates Adv'. In particular, if the attack results in breaking the secret key sk_i , then Adv also obtains sk_i .

3.4 Possession of sk_1 and sk_2

Unforgeability does not preclude a situation in which devices dev_1, dev_2 holding, respectively, sk_1, sk_2 execute a two-party computation whose final output is a kleptographic signature (R_x, s) . However, such execution would require additional communication, which could be noticed by the device holders. We assume that no additional communication can take place! In this situation, the entire computation that leads to the derivation of (R_x, s) must take place on a single device. But then, by the unforgeability property, this device must hold both sk_1 and sk_2 .

Conclusions

The construction from Section 2.2 serves merely as a proof-of-concept. Note that we do not need sophisticated indistinguishability compared to the original protocol (for example, in terms of undetectability via side-channel analysis), as the kleptographic proof may be implemented within the scope of activities planned by the EU Commission's Roadmap [8], where indistinguishability might be broken by assumption. Without going into a discussion about the pros and cons of these plans, our primary goal is to show technical feasibility within the currently planned EDIW architecture. Note that not only unlinkability is broken, but a proof can be presented to third parties, e.g., in a court, so it may serve legal purposes.

On the other hand, we cannot ignore the threat that pseudonyms may be generated in the manner described above and only appear to protect user's privacy. Consequently, we believe that this threat should be addressed in the certification of EDIW solutions.

Acknowledgment The research of the third author was partially funded by the NCN National Science Centre, Poland under the OPUS call in the Weave programme NCN 2023/51/I/ST6/02770. The third and fourth authors thank the Polish Ministry of Digitalization for support in developing their competences about the eIDAS Regulation.

References

1. The European Parliament and the Council of the European Union: Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (GDPR). Official Journal of the European Union **L 119/1** (2016)
2. The European Parliament and the Council of the European Union: Regulation (EU) No 910/2014 on electronic identification and trust services for electronic transactions in the internal market (eIDAS Regulation). Official Journal of the European Union **L 257/73** (2014)
3. The European Parliament and the Council of the European Union: Regulation (EU) No 2024/1183 amending Regulation (EU) no 910/2014 as regards establishing the European Digital Identity Framework. Official Journal of the European Union **L series** (2024)
4. European Commission: Architecture and Reference Framework (ARF) for the EU Digital Identity Wallet. Available at: <https://eudi.dev/3.0.0/about/> (started in 2023)
5. European Commission: Commission Implementing Regulation (EU) 2024/2979 laying down rules as regards the integrity and core functionalities of European Digital Identity Wallets. Official Journal of the European Union **L series** (2024)
6. World Wide Web Consortium: Web authentication: An API for accessing public key credentials level 2. <https://www.w3.org/TR/2021/REC-webauthn-2-20210408> (2021)
7. European Commission: The GitHub repository for the Architecture and Reference Framework (ARF). Available at: <https://github.com/eu-digital-identity-wallet/eudi-doc-architecture-and-reference-framework/> (repository active in 2026)
8. European Commission: Communication on a Roadmap for lawful and effective access to data for law enforcement. <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52025DC0349> (2025) COM(2025) 349 final.
9. Kutylowski, M., Lauks-Dutka, A., Kubiak, P., Zawada, M.: FIDO2 facing kleptographic threats by-design. *Applied Sciences* **14**(23) (2024)
10. Bundesamt für Sicherheit in der Informationstechnik: Technical Guideline TR-03110-2: Advanced Security Mechanisms for Machine Readable Travel Documents and eIDAS Token – Part 2. Technical recommendation (2016) Version 2.21.
11. Błażkiewicz, P., Kutylowski, M., Słowik, M.: Signature-in-signature: the last line of defense in case of signing key compromise. In: *Computer Security – ESORICS 2021*. Volume 12973 of LNCS., Springer (2021) 777–782
12. Dayanikli, D., Lang, J., Lehmann, A.: Signatures with post-compromise accountability. *IACR Cryptol. ePrint Arch.* (2026) 1203. Available from: <https://eprint.iacr.org/2026/1203>
13. Bundesamt für Sicherheit in der Informationstechnik: Elliptic Curve Cryptography. Technical recommendation (2018) Version 2.10.
14. Pointcheval, D., Stern, J.: Security arguments for digital signatures and blind signatures. *Journal of Cryptology* **13** (2001)
15. Goyal, V., O’Neill, A., Rao, V.: Correlated-input secure hash functions. In Ishai, Y., ed.: *Theory of Cryptography TCC Proceedings*. Volume 6597 of LNCS., Springer (2011) 182–200

On the Applicability of Traffic-splitting Strategies as a Protection Shield against End-to-end Traffic Correlation Attacks in Tor

Asya Mitseva¹, Bozhan Cai¹, and Andriy Panchenko¹

Brandenburg University of Technology, 03046 Cottbus, Germany
{asya.mitseva, bozhan.cai, andriy.panchenko}@b-tu.de
<https://www.b-tu.de/en/fg-it-sicherheit>

Abstract. Passive end-to-end (E2E) traffic correlation against Tor users is a special type of traffic analysis attack that aims to link traffic segments of a user’s connection entering and exiting the Tor network. Through outside the attacker’s model in Tor, previously such an attack was also considered impractical due to the required power of the attacker to observe both ends of a Tor user connection at the same time. Hence, research on defenses against E2E traffic correlation has received little attention over the years. However, the efficacy of the E2E traffic correlation attacks has significantly evolved over the past decade, prompting questions about the validity of these assumptions.

Concurrently, a major line of research has focused on designing defenses against website fingerprinting (WFP)—another type of passive analysis attack aiming to reveal the content of encrypted and anonymized connections (i.e., the visited website) by analyzing patterns of data flows exchanged between a Tor user and its first anonymization node. Despite the similarity in executing WFP and E2E traffic correlation, research aiming to design efficient and practical defenses against these attacks is distinct. This work aims to bridge this gap by assessing the effectiveness of two state-of-the-art WFP defenses based on traffic splitting against E2E traffic correlation. Our results show that these defenses are not only efficient against WFP but also against modern E2E traffic correlation classifiers, significantly reducing their detection rates and rendering them impractical in real world. Our work provides a glimmer of hope that the realization of a countermeasure that protects against two prevalent traffic analysis attacks is possible.

Keywords: Traffic Analysis · End-to-end traffic correlation · Tor.

1 Introduction

Internet-based digital technologies, such as the World Wide Web, enable individuals to access an abundance of information and exercise their freedom of speech. However, Internet users leave personal digital traces with each online action, which can be exploited for marketing purposes, user tracking, and even

controlling their online access and activities. While encryption helps protect the confidentiality of user data exchanged over the Internet, it does not obfuscate the metadata associated with the communication, such as packet sizes, direction, timing, and the relationship between both communicating parties. To help affected users reduce the amount of information leakage generated by their Internet usage, Tor [5] is currently the most popular anonymization network, designed particularly for low-latency applications. Tor employs the onion-routing principle, whereby user traffic is encrypted in multiple layers and transmitted over a chain of three Tor nodes, called entry, middle, and exit. Since each Tor node can remove (or add) only a single layer of encryption, no intermediate entity observing the communication between the Tor user and its destination can know the identities (i.e., IP addresses) of both communicating parties at the same time.

Despite Tor’s promise to hide the relationship between the user and its destination from a *local passive* observer, e.g., an Internet service provider (ISP) or a malicious Tor node, it cannot conceal the number, direction, and timing of transmitted packets. This information can be exploited for sophisticated attacks, such as website fingerprinting (WFP) and end-to-end (E2E) traffic correlation. WFP aims to identify the content of encrypted and anonymized connections (i.e., the visited website) by observing and analyzing patterns of data flows exchanged between a Tor user and its entry node. In E2E traffic correlation, an attacker can observe both the traffic transmitted between the Tor user and its entry node as well as the traffic between the exit node and the user’s destination. The goal of the adversary is to determine whether both traffic segments correspond to a single Tor user connection, thereby revealing their relationship. Unlike E2E traffic correlation, WFP is covered by the Tor’s threat model, which resulted in a plethora of WFP countermeasures proposed over the last decade [17].

Although E2E traffic correlation is explicitly excluded from the Tor’s threat model, such an attack was previously considered impractical. On the one hand, multiple factors cause significant differences between the ingress and egress Tor traffic, such as multiplexing of encrypted traffic (i.e., sending multiple independent application streams over the same Tor connection), the use of fixed-size data units (i.e., cells) between Tor nodes, delays created by the Tor’s flow control mechanism, and network indeterminism [25]. On the other hand, E2E traffic correlation requires more computational resources than WFP due to its pairwise nature, which necessitates comparing all possible pairs of traffic to deanonymize a given set of users. Given the increased number of volunteer Tor nodes and ISPs carrying Tor traffic over the years, the natural assumption was that the likelihood of a single entity being able to observe user traffic entering and exiting Tor had diminished [5]. However, the significant evolution of E2E traffic correlation attack efficacy over the past decade, coupled with the analysis of the susceptibility of Tor users to network-level adversaries, such as autonomous systems (ASs) and Internet eXchange Points (IXPs)¹, prompt questions about the validity of this assumption [22, 25, 34]. Previous research [20] has shown that an attacker positioned at an IXP can observe a significant amount of Tor traffic and corre-

¹ A shared physical infrastructure in a single location connecting several ASs.

late high-speed network flows at low sampling rates by observing traffic passing through any AS co-located at that IXP, thereby compromising users’ anonymity. Even worse, an attacker possessing multiple ASs or IXPs can compromise Tor connections at a significantly faster rate [11].

Despite the similarity in executing both WFP and E2E traffic correlation, research on designing efficient and practical defenses against these attacks has been independent of each other, leading to little attention being given to E2E traffic correlation defenses over the years. Concurrently, a major line of research has focused on the design of WFP defenses that rely on various defensive mechanisms, such as constant or adaptive padding, traffic morphing, traffic splitting, and adversarial perturbations, some of which show themselves as attractive candidates for adoption in Tor [17]. In this work, we aim to bridge the gap between these communities and assess the effectiveness of two state-of-the-art WFP defenses, called TrafficSliver [3], based on traffic splitting against E2E traffic correlation. Our selection is based on the fact that these defenses cause practically no overhead while being highly efficient against WFP attacks [17]. Our results show that these defenses are efficient not only against WFP, but also against modern E2E traffic correlation classifiers, such as DeepCorr [22] and DeepCoFFEA [25], rendering them impractical in real world. Our work provides a glimmer of hope that a countermeasure protecting against the two prevalent traffic analysis attacks, WFP and E2E traffic correlation, is practically possible with low overhead.

2 Related Work

This section reviews the operational aspects of previous state-of-the-art E2E traffic correlation attacks and defenses. E2E traffic correlation is typically classified into passive and active methods [1, 22]. Similar to prior work [22, 25], we omit the discussion of active E2E traffic correlation methods, as such attacks are easily detectable and, thus, cannot be deployed by an attacker on a large scale for a long time period without being detected. Moreover, we briefly discuss prior work aiming to increase the attacker’s chances of being located at both ends of Tor—a mandatory requirement for performing E2E traffic correlation—and existing defenses against such attacks.

E2E traffic correlation attacks. Early E2E traffic correlation methods [10, 26, 38] relied on general statistical correlation metrics to quantify the degree of similarity between the traffic entering and exiting the anonymization network. However, these methods require unrealistically long traffic traces to make reliable decisions. Other approaches [15, 31] divided the monitored traffic at both ends of the anonymization network into fixed-size time windows and correlated the times between packet arrivals and the number of packets transmitted in each window. Nasr et al. [21] aimed to improve the scalability of E2E traffic correlation using linear projection algorithms to compress the requisite data, and quantify the degree of similarity between traffic entering and exiting Tor using cosine similarity. Sun et al. [34] proposed an asymmetric E2E traffic correlation approach, wherein the attacker only needs to observe paths in different direc-

tions at both ends of Tor. The authors show constant correlation dependence between metadata of data traffic collected from one side of Tor and the TCP sequence number field and TCP acknowledgment numbers observed on the other side. Despite these advances, the main drawbacks of these methods remain their high false positive rate (FPR) and the need for impractically long traffic traces to make reliable correlations.

Nasr et al. [22] also proposed DeepCorr—the first deep learning (DL) based E2E traffic correlation attack. DeepCorr is based on a Convolutional Neural Network with a correlation function that is tailored specifically to link traces generated by arbitrary Tor users to arbitrary destinations. It takes as an input both the inter-arrival packet times and the packet sizes of the traffic transmitted at both ends of Tor. DeepCorr achieves significantly higher detection rates in correlating Tor traffic compared to methods using general statistical correlation metrics, while requiring substantially shorter traffic traces than prior work. AttCorr [16] is another DL-based E2E traffic correlation attack, based on a transformer model. However, the authors of AttCorr were unable to achieve a superior detection rate compared to that reported in [22]. Rimmer et al. [30] criticized the manner in which E2E traffic correlation attacks are evaluated in research. The authors argued that the typical experimental setup used in previous studies yields biased results that do not align with actual real-world conditions and artificially enhances the attack’s detection rate. The authors show that using a single proxy to collect network traffic exiting the Tor network often introduces latency that benefits the attacker, which is uncommon in real-world scenarios.

Oh et al. [25] identified further limitations in the scalability of prior E2E traffic correlation methods, such as DeepCorr. The authors highlighted their computational complexity due to their pairwise nature and the high number of falsely correlated traces still produced. To address these issues, they proposed DeepCoFFEA, a new E2E traffic correlation approach based on feature embedding networks and amplification. DeepCoFFEA uses two different DL-based models, one for ingress traffic and one for egress traffic, to reduce the dimensionality of the input traffic traces. The reduced data is then utilized for correlation computations. To mitigate miscorrelations during these computations, DeepCoFFEA divides the input traffic traces into shorter, time-based segments and employs a voting process on this set to make a final decision. Another E2E traffic correlation technique is FlowTracker [8], which relies on a cumulative representation of observed traffic and employs stacked autoencoders to reduce the negative impact of communication noise. The authors use information from unrelated traffic traces to generate differential representations that amplify the essential differences between similar but unrelated traces, thus reducing the confusion caused by large candidate trace sets.

E2E traffic correlation defenses have received little attention. Early proposals indicated the potential for employing different padding strategies and inserting dummy packets [15, 31]. However, these countermeasures have not been implemented in practice due to their considerable bandwidth and latency overhead and reduced protection against modern E2E traffic correlation attacks [25]. Nasr et al. [22] showed that Tor pluggable transports—techniques, designed to

obfuscate potential distinctions between Tor- and non-Tor-protected traffic—that introduce padding to obfuscate packet sizes and randomize inter-arrival packet times can significantly reduce the detection rate of DeepCorr. However, Oh et al. [25] showed that neither Tor pluggable transports nor low-cost WFP defenses (WTF-PAD [13] and FRONT [7]) can adequately protect against DeepCoFFEA. Recent work [9, 23] on WFP defenses has speculated that their methods could also serve to protect against E2E traffic correlation. However, these proposals lack concrete empirical results and are still subject to adoption issues, which hinders their deployment in practice. Since defenses based on traffic splitting provide the greatest protection against WFP [17], this study examines the effectiveness of two such defenses against state-of-the-art E2E traffic correlation attacks and suggests ways to improve their splitting strategies.

Attacker’s placement for E2E traffic correlation. Executing E2E traffic correlation requires the adversary to have access to favorable vantage points to observe relevant network traffic. Two main groups of attackers can employ different strategies to place themselves on the path between the Tor user and its destination: *network-level entities*, e.g., malicious ASs and IXPs, and *malicious Tor nodes*. Early research [6, 18, 37] focused primarily on analyzing the susceptibility of Tor users to ASs that naturally observe both ends of Tor user connections simultaneously. Murdoch and Zieliński [20] showed that an attacker positioned at an IXP can correlate high-speed network flows by observing traffic passing through any AS co-located at that IXP, at low sampling rates. Johnson et al. [11] explored the time required for an attacker to compromise Tor circuits when controlling a set of ASs or IXPs. Following the introduction of asymmetric E2E traffic correlation [34], other studies [24, 35, 36] reevaluated the threat posed by malicious ASs and IXPs, and their potential for employing active traffic redirection attacks against Border Gateway Protocol (BGP). Conversely, malicious Tor nodes exploit weaknesses in Tor’s bandwidth measurement system to create the impression of greater bandwidth capacity, thus increasing their selection by Tor users [2, 12, 32, 33]. In response, some research has focused on designing alternative Tor path selection methods [24, 35] aiming to limit the likelihood that specific ASs and IXPs appear at both ends of a Tor user’s connection, as well as improved bandwidth measurement systems [12] aiming to minimize the impact of malicious nodes. Unlike these studies, whose goal is to limit the proactive placement of attackers in suitable locations for E2E traffic correlation, this work analyzes defenses that aim to thwart E2E traffic correlation itself.

3 Traffic Splitting against Traffic Correlation

The main objective of this work is to evaluate the effectiveness of two WFP defenses against E2E traffic correlation attacks that provide the highest level of protection against WFP with the lowest overhead. As shown by Mathews et al. [17], the most notable group of defenses serving as candidates for adoption in Tor are those that distribute user traffic over multiple Tor paths. This group of defenses is notable for its high level of protection against WFP, as well as its effi-

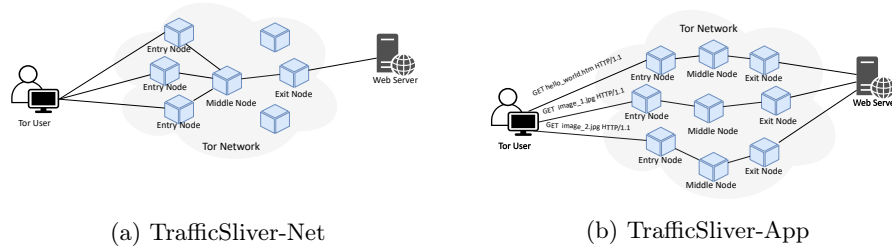


Fig. 1: An overview of the operation of both TrafficSliver defenses.

ciency in terms of bandwidth and latency overhead. Both TrafficSliver defenses, TrafficSliver-Net and TrafficSliver-App, which were proposed in [3], currently provide the most optimal balance between the level of protection against WFP and the bandwidth and latency overheads introduced in Tor. Therefore, in this work we focus on analyzing their robustness against E2E traffic correlation.

Figure 1 shows the operation of both TrafficSliver defenses. TrafficSliver-Net (Figure 1a) is a network-layer defense that employs the concept of multipathing within Tor. In other words, the user traffic is sent via several Tor paths that contain distinct entry nodes but share the same middle and exit nodes. To this end, the defense is deployed on the user’s onion proxy (OP)² and the Tor middle node, which, in turn, implement mechanisms to distribute and reassemble user traffic from multiple entry nodes. TrafficSliver-Net applies a batched weighted random splitting strategy, in which a randomly generated list of weights is used to select an entry node for a predefined batch of Tor cells to be transmitted.

Unlike TrafficSliver-Net, TrafficSliver-App (Figure 1b) operates completely independent of the underlying anonymization network and does not require the cooperation of Tor nodes to split traffic. It acts as a local proxy between the user’s browser and OP and creates multiple Tor circuits, each with a unique entry node. Then, it randomly sends separate full HTTP requests for distinct web objects belonging to the same website over different entry nodes. Unlike TrafficSliver-Net, the initial implementation of TrafficSliver-App does not restrict users from using the same middle and exit nodes for a single Tor user connection. In this work, we expand the operational scope of TrafficSliver-App by introducing two additional modes. In particular, we instruct a Tor user to employ either (i) different exit nodes or (ii) the same exit node for its multipath circuits. In the former case, the number of exit nodes corresponds to the number of unique entry nodes used for the user’s multipath connection. The goal of this extension is to analyze how using different exit nodes impacts the level of protection that TrafficSliver-App provides against E2E traffic correlation.

4 Experimental Setup

To evaluate the level of security that both TrafficSliver defenses provide against E2E traffic correlation, we need to simultaneously collect traffic at two vantage

² A local software that runs on user’s machine and directs user’s traffic through Tor.

points: the traffic exchanged between a Tor user and the entry nodes, and the traffic transmitted between the exit nodes and the final destination. To collect ingress Tor traffic, we deploy our own Tor users and sniff the traffic exchanged with the Tor network. To collect traffic transmitted between a Tor exit node and the intended destination, we configure a proxy server to which we direct traffic from our deployed Tor users before it reaches its final destination, similar to prior work [22, 30, 25]. This is needed due to the absence of provisions in the Tor protocol that allow users to instruct exit nodes to tunnel traffic through an alternate proxy before delivering the data. While the preparation of such an experimental setup for TrafficSliver-Net is rather straightforward, TrafficSliver-App necessitates multiple design choices that require careful analysis. From a technical perspective, to force Tor user traffic to traverse an additional proxy after the exit node implies the implementation of an extra layer of encapsulation. This poses significant challenges in the context of TrafficSliver-App since it functions as an additional HTTP(S) proxy for the Tor user.

Datasets. Analogous to previous work [22, 30, 25], we rely on web browsing based network traffic generated by our deployed Tor users to evaluate the efficiency of both TrafficSliver defenses against E2E traffic correlation. Although E2E traffic correlation can be applied to any TCP-based user traffic transmitted via Tor, the goal of this work is to analyze the effectiveness of two existing WFP defenses against this attack necessitating the consideration of web browsing. To this end, we compiled a new dataset consisting of the index pages of 500 unique websites from the latest Tranco Top list of the most popular websites [14]. We visited each of these websites once and collected the corresponding pairs of ingress and egress traffic traces, with each of these traces consisting of at least 300 packets, as recommended in [22].

Network traffic collection. For each website, we collect the underlying network traffic transmitted between our own deployed Tor users and the corresponding Tor entry nodes, as well as between the corresponding Tor exit nodes and our users’ final destinations. To avoid any additional modifications to our deployed Tor users, which could affect the operation of both TrafficSliver defenses, especially TrafficSliver-App, we setup our own HTTP(S) proxy server to collect traffic transmitted between Tor exit nodes and the final destinations. This proxy acts as a reverse proxy using mitmproxy [4], which is dynamically reconfigured by our automated tools for each new website visited by our Tor users. Since the reverse proxy for each of our websites is active for a few seconds to load the corresponding page, we believe that it cannot attract any other user traffic or temporarily block connections to particular websites. In addition, we automatically reject user requests that do not originate from the Tor network to reduce the operational load on our proxy server. The proxy is located within the university campus network, whose IP address and respective port are publicly available for connections from the Internet. We further equipped the proxy with a set of automated tools to sniff traffic passing through it and capture the timestamps needed for processing that traffic subsequently.

We used the automated approach presented in [27] with Tor Browser 13.5.2 to collect metadata on ingress Tor traffic for all websites in our dataset, includ-

ing TCP packet sizes, direction, and timing. We extended this approach further to redirect our Tor users’ traffic to traverse the HTTP(S) proxy server prior to reaching their final destinations. Contrary to [19, 27], we considered the TCP packets rather than Tor cells, due to the absence of Tor-protected network traffic between the Tor exit nodes and our HTTP(S) proxy server. Similar to related work [19], we excluded page loads denying requests coming from Tor, showing a CAPTCHA, having no content, or pointing to a client or server error. The attacker is not interested in fingerprinting such page loads since none of them can be considered successful in accessing the content. Furthermore, we launched a middle node in the real Tor network that was used by our deployed Tor users to perform traffic splitting and merging, as required by TrafficSliver-Net [3]. For both TrafficSliver defenses, we selected the corresponding best traffic-splitting strategy, as outlined in [3], the only difference being that the best traffic-splitting strategy for TrafficSliver-App was evaluated using the additional modes, presented in Section 3.

Classifiers and evaluation setup. For our evaluation, we consider two state-of-the-art E2E traffic correlation classifiers, DeepCorr [22] and DeepCoF-FEA [25]. We refer the reader to Section 2 and the original papers for more details about their operation. In all subsequent experiments, unless a different methodology is explicitly mentioned, we apply 5-fold cross-validation (CV) with respect to the total number of collected pairs of ingress and egress traffic traces. Similar to related work [22, 30, 25], we compute the true positive rate (TPR), measuring the fraction of correlated pairs that were correctly detected, and the FPR, expressing the fraction of pairs that were falsely identified as correlated.

5 Evaluation and Discussion

In this section, we evaluate the effectiveness of both TrafficSliver defenses against both state-of-the-art E2E traffic correlation attacks. We assume that the attacker controls one malicious entry node and one malicious exit node in Tor user’s multipath connections. For all experiments, we assume that the attacker is aware of the applied defense, including its splitting scheme. We also ensure that all multipath traces of the same page load are used either for testing or for training only. In addition, we performed a series of experiments to first identify the optimal hyperparameters and threshold for both E2E traffic correlation classifiers. Due to space constraints, we only report the best classification results in the rest of the section. We further assume that the attacker has sufficient resources to collect a representative training dataset. When testing the classifiers, we always utilized n positively correlated trace pairs³ in each CV iteration and created $(n - 1)$ negatively correlated trace pairs for each positively correlated pair. For each E2E traffic correlation classifier, we first conducted an experiment using non-defended Tor traffic to establish a baseline. Additionally, we compared the detection rates obtained for both TrafficSliver defenses with those achieved for two state-of-the-art Tor pluggable transports.

³ Ingress and egress traffic segments corresponding to the same Tor user’s connection.

TrafficSliver defenses against E2E traffic correlation. Figure 2 illustrates the classification results obtained for both E2E traffic correlation classifiers. In case of DeepCorr (Figure 2a), the TPR is cut in at least half for the same FPR when both TrafficSliver defenses are applied. Although TrafficSliver-Net achieves the lowest TPR for a given FPR, we observe that the difference in detection rate between TrafficSliver-Net and TrafficSliver-App with a single exit node is negligible. One possible reason for the slightly higher classification results for TrafficSliver-App with multiple exit nodes is the frequent selection of the same entry and exit nodes as a positive pair, over which the same ingress and egress traffic segments of the same part of the multipath connection were transmitted. Since the TrafficSliver-App requests different objects through different circuits, this contributes to their isolation and, if the objects are unique enough, yields better classification results than a single exit, where the objects are not separated from the perspective of an external observer.

In case of DeepCoFFEA (Figure 2b), we observe an even larger drop in the classifier’s detection rate for both TrafficSliver defenses. This result is surprising, given that DeepCoFFEA is considered to be a more powerful attack than DeepCorr. This finding underlines the hypothesis that the most effective attack in vanilla settings does not necessarily imply superiority in the presence of a defense. TrafficSliver-App achieves a slightly higher TPR for the same FPR than TrafficSliver-Net, regardless of whether a single or multiple exit nodes are used. However, the detection rates are low enough to conclude that DeepCoFFEA is practically infeasible when either TrafficSliver defense is employed. We also notice that, for a low FPR, the difference in TPR when using a single versus multiple exit nodes for TrafficSliver-App is negligible. I.e., knowledge of either single objects or the whole stream does not provide significant information here. Since a single exit node is sufficient for defense purposes, there is no need to use several and increase the load on Tor’s limited number of exit nodes. While Oh et al. [25] showed the DeepCoFFEA’s success against two other modern WFP defenses, WTF-PAD and FRONT, related work on WFP [3, 19] has shown that these defenses do not provide sufficient level of protection in other settings.

Comparison with Tor pluggable transports. Similar to prior work [22, 25], we compare the detection rates obtained for both TrafficSliver defenses to those obtained for two state-of-the-art Tor pluggable transports Obfs4 [28] and Snowflake [29]. As shown in Figure 2, Obfs4 achieves a slightly lower detection rate than TrafficSliver-Net for DeepCorr. However, its protection level significantly decreases compared to both TrafficSliver defenses in case of DeepCoFFEA, rendering Obfs4 ineffective in practice. Conversely, we observe that Snowflake achieves consistently low detection rates for both E2E traffic correlation classifiers, nearly equivalent to random guessing. Although these results indicate a higher protection level for Tor users than with both TrafficSliver defenses, the operation of Snowflake relies on domain fronting and is heavily dependent on the cooperation of Content Delivery Networks (CDNs), which makes its long-term existence questionable. Moreover, the small number of Tor bridges (i.e., Tor nodes that are not publicly announced to avoid blocking their IP addresses) sup-

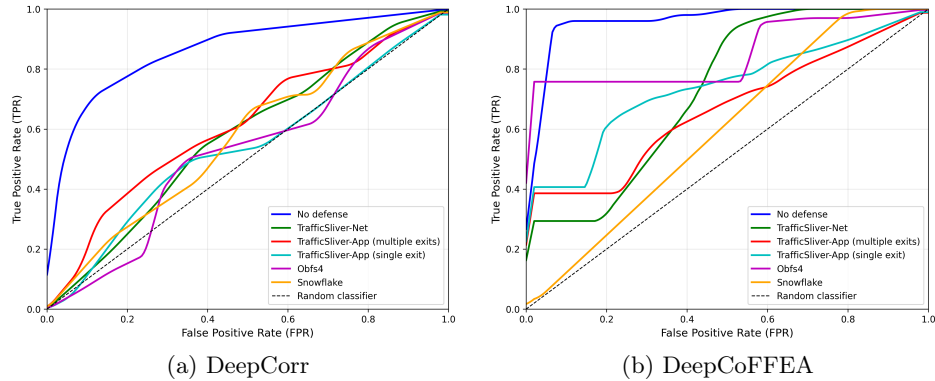


Fig. 2: Detection rate of both state-of-the-art E2E traffic correlation classifiers achieved in different scenarios.

porting Obfs4 and Snowflake, and the higher latency and bandwidth overhead these pluggable transports generate, render them impractical for general use by all Tor users. To sum up, countermeasures based on traffic splitting are able to defend against both WFP and E2E traffic correlation attacks with only moderate performance overhead without depending on CDNs and scale gracefully.

Discussion and limitations. We rely on a single proxy server to collect the traffic exchanged between Tor exit nodes and final destinations. As shown in [30], using a single proxy for data collection introduces artificial time patterns due to the increased latency required to load a website. This could bias our evaluation results and allow the attacker to achieve a higher TPR at a lower FPR than in reality. However, since our main goal is to evaluate the security level provided by WFP defenses, not to propose a new E2E traffic correlation attack, potential biases in our evaluation results may also imply that both TrafficSliver defenses would provide an even higher level of protection against E2E traffic correlation in reality.

6 Conclusion

This work analyzed the suitability of traffic splitting based WFP defenses to protect against E2E traffic correlation. This analysis is a fundamental step on advancing the adoption of effective WFP countermeasures in the real Tor network. WFP defenses that provide protection against other passive deanonymization attacks, besides offering a high level of protection against WFP and low latency and bandwidth overhead, are more attractive for real-world deployment. Our evaluation results showed that two state-of-the-art WFP defenses based on traffic splitting are able to dramatically reduce the detection rate of modern E2E traffic correlation classifiers. Although one of the pluggable transports provided a higher security level, it relies on domain fronting and requires the use of a Tor bridge, which are not applicable to general usage by all Tor users. Our evaluation provides valuable insights towards the selection of effective countermeasure for hampering popular traffic analysis attacks in Tor that can be deployed at scale.

Acknowledgments. This research is supported by the German Federal Government, the Federal Ministry of Research, Technology and Space and the State of Brandenburg within the framework of the joint project EIZ: Energy Innovation Center (project numbers 85056897 and 03SF0693A) with funds from the Structural Development Act (Strukturstärkungsgesetz) for coal-mining regions.

References

1. Alsabab, M., Goldberg, I.: Performance and security improvements for tor: A survey. *ACM Computing Surveys* **49**(2) (2016)
2. Bauer, K., et al.: Low-resource Routing Attacks Against Tor. In: Workshop on Privacy in Electronic Society. ACM (2007)
3. De la Cadena, W., et al.: Trafficsliver: Fighting website fingerprinting attacks with traffic splitting. In: SIGSAC Conference on Computer and Communications Security. CCS, ACM (2020)
4. Cortesi, A., et al.: Mitmproxy. <https://mitmproxy.org/> (2026)
5. Dingledine, R., et al.: Tor: The Second-Generation Onion Router. In: 13th Conference on USENIX Security Symposium. USENIX Association (2004)
6. Edman, M., Syverson, P.: AS-awareness in Tor Path Selection. In: 16th Conference on Computer and Communications Security. CCS, ACM (2009)
7. Gong, J., Wang, T.: Zero-delay Lightweight Defenses against Website Fingerprinting. In: 29th USENIX Security Symposium. USENIX Association (2020)
8. Guan, Z., et al.: FlowTracker: Improved flow correlation attacks with denoising and contrastive learning. *Computers & Security* **125** (2023)
9. Holland, J.K., et al.: DeTorrent: An Adversarial Padding-only Traffic Analysis Defense. *Proceedings on Privacy Enhancing Technologies* **2024**(1) (2024)
10. Jin, J., Wang, X.: On the effectiveness of low latency anonymous network in the presence of timing attack. In: IEEE/IFIP International Conference on Dependable Systems & Networks. IEEE (2009)
11. Johnson, A., et al.: Users Get Routed: Traffic Correlation on Tor by Realistic Adversaries. In: Conference on Computer & Communications Security. CCS, ACM (2013)
12. Johnson, A., et al.: PeerFlow: Secure Load Balancing in Tor. 17th Symposium on Privacy Enhancing Technologies (2017)
13. Juarez, M., et al.: Toward an Efficient Website Fingerprinting Defense. In: Proceedings of the 21st European Symposium on Research in Computer Security. ESORICS, Springer (2016)
14. Le Pochat, V., et al.: Tranco: A Research-Oriented Top Sites Ranking Hardened Against Manipulation. In: 26th Annual Network and Distributed System Security Symposium. NDSS, Internet Society (2019)
15. Levine, B.N., et al.: Timing Attacks in Low-Latency Mix Systems. In: Financial Cryptography. FC, Springer (2004)
16. Li, J., et al.: AttCorr: A Novel Deep Learning Model for Flow Correlation Attacks on Tor. In: International Conference on Consumer Electronics and Computer Engineering. ICCECE, IEEE (2021)
17. Mathews, N., et al.: Sok: A critical evaluation of efficient website fingerprinting defenses. In: Symposium on Security and Privacy. S&P, IEEE (2023)
18. Mathewson, N., Dingledine, R.: Location Diversity in Anonymity Networks. In: Workshop on Privacy in the Electronic Society. WPES, ACM (2004)

19. Mitseva, A., Panchenko, A.: Stop, Don't Click Here Anymore: Boosting Website Fingerprinting By Considering Sets of Subpages. In: 33rd USENIX Security Symposium. USENIX Association (2024)
20. Murdoch, S.J., Zielinski, P.: Sampled Traffic Analysis by Internet-exchange-level Adversaries. In: Privacy Enhancing Technologies. PETS, Springer (2007)
21. Nasr, M., et al.: Compressive Traffic Analysis: A New Paradigm for Scalable Traffic Analysis. In: SIGSAC Conference on Computer and Communications Security. CCS, ACM (2017)
22. Nasr, M., et al.: DeepCorr: Strong Flow Correlation Attacks on Tor Using Deep Learning. In: SIGSAC Conference on Computer and Communications Security. CCS, ACM (2018)
23. Nasr, M., et al.: Defeating DNN-Based Traffic Analysis Systems in Real-Time with Blind Adversarial Perturbations. In: 30th USENIX Security Symposium. USENIX Security (2021)
24. Nithyanand, R., et al.: Measuring and Mitigating AS-level Adversaries Against Tor. In: 23th Annual Network & Distributed System Security Symposium. NDSS, Internet Society (2016)
25. Oh, S.E., et al.: DeepCoFFEA: Improved Flow Correlation Attacks on Tor via Metric Learning and Amplification. In: Symposium on Security and Privacy. S&P, IEEE (2022)
26. Palmieri, F.: A Distributed Flow Correlation Attack to Anonymizing Overlay Networks Based on Wavelet Multi-Resolution Analysis. *IEEE Transactions on Dependable and Secure Computing* **18**(5) (2021)
27. Panchenko, A., et al.: Website Fingerprinting at Internet Scale. In: 23rd Annual Network and Distributed System Security Symposium. NDSS, Internet Society (2016)
28. Project, T.T.: Tor at the Heart: Bridges and Pluggable Transports. <https://blog.torproject.org/tor-heart-bridges-and-pluggable-transports/> (2016)
29. Project, T.T.: Free Your Connection. <https://snowflake.torproject.org/> (2026)
30. Rimmer, V., et al.: Trace Oddity: Methodologies for Data-Driven Traffic Analysis on Tor. *Proceedings on Privacy Enhancing Technologies* **2022**(3) (2022)
31. Shmatikov, V., Wang, M.H.: Timing Analysis in Low-Latency Mix Networks: Attacks and Defenses. In: *Computer Security – ESORICS 2006*. Springer (2006)
32. Snader, R., Borisov, N.: A Tune-up for Tor: Improving Security and Performance in the Tor Network. In: 16th Network and Distributed System Security Symposium. Internet Society (2008)
33. Snader, R., Borisov, N.: Improving Security and Performance in the Tor Network through Tunable Path Selection. *IEEE Transactions on Dependable and Secure Computing* **8**(5) (2011)
34. Sun, Y., et al.: RAPTOR: Routing Attacks on Privacy in Tor. In: 24th USENIX Security Symposium. USENIX Association (2015)
35. Tan, H., et al.: Data-plane Defenses against Routing Attacks on Tor. *Privacy Enhancing Technologies* **4** (2016)
36. Vanbever, L., et al.: Anonymity on QuickSand: Using BGP to Compromise Tor. In: 13th Workshop on Hot Topics in Networks. HotNets, ACM (2014)
37. Wacek, C., et al.: An Empirical Evaluation of Relay Selection in Tor. In: 20th Annual Network & Distributed System Security Symposium. NDSS, Internet Society (2013)
38. Zhu, Y., et al.: Correlation-Based Traffic Analysis Attacks on Anonymity Networks. *IEEE Transactions on Parallel and Distributed Systems* **21**(7) (2010)

AFTERNOON SESSION 2

Applied and Regulatory Privacy

From Static Schedules to Consent-Driven Retention: An Approach to GDPR Storage Limitation Enforcement

Samuel Opoku Daniels¹ and Haralambos Mouratidis¹

Institute for Analytics and Data Science,
School of Computer Science and Electronic Engineering, University of Essex, UK
{s.daniels,h.mouratidis}@essex.ac.uk

Abstract. The General Data Protection Regulation (GDPR) Article 5(1)(e) mandates that personal data be retained only as long as necessary for its processing purpose (storage limitation), yet current privacy management systems implement this as static retention schedules disconnected from the consent state that authorises the processing. When consent is withdrawn or a purpose is completed, the retention obligation must be reassessed, but existing approaches lack mechanisms to enforce these dynamic dependencies. This paper presents a consent-derived retention enforcement mechanism that formally binds retention periods to processing purposes and consent lifecycle states. It introduces three contributions: (i) a *Retention Policy Model* linking retention durations to consent conditions with jurisdictional overrides; (ii) *purpose-linked erasure rules* specifying how data transitions through active, expired, held, and erased states under consent withdrawal, purpose completion, and legal hold (Art. 17(3)); and (iii) *automated compliance constraints* encoding storage limitation invariants traceable to GDPR Articles 5(1)(e), 17, and 25. The mechanism extends ConsentML within the DIAPROD framework and is implemented in the DIAPROT modelling tool. A UK open banking case study across six scenarios shows that consent-derived enforcement detects violations that static schedules miss.

Keywords: Data Privacy Management · Storage Limitation · GDPR Compliance · Consent Lifecycle · Retention Enforcement · Right to Erasure · Data Protection by Design

1 Introduction

The idea of storage limitation is one of the seven fundamental tenets of the General Data Protection Regulation (GDPR) [8]. Article 5(1)(e) mandates that personal data be retained in a manner that allows for the identification of data subjects only for as long as is necessary for the purposes of processing their personal data [1]. Article 17(1)(a) confers the right to erasure when data is no longer required for its initial purpose [1]. Article 25 emphasises the need for data protection to be included by design and by default (PbD) [1].

Over 30% of GDPR articles relate to storage [21]; as a result, a significant number of fines [19] have already been issued to companies found to be in breach of this legal requirement. In February 2026, the European Data Protection Board (EDPB) released the findings of its 2025 coordinated enforcement initiative, in which 32 Data Protection Authorities evaluated 764 data controllers across Europe on erasure practices, revealing ongoing deficiencies in systematic data classification and automated deletion [13]. In 2025, the EDPB initiated a new Coordinated Enforcement Framework that explicitly addresses the right to erasure (Art. 17), affirming its prominence as one of the most frequently invoked GDPR rights [13].

Despite significant regulatory pressures, a core disconnect persists between what the law stipulates and how it is implemented in privacy management practice. In current applications, storage limitations are often treated as a fixed retention schedule: a compliance officer establishes specific retention periods (e.g., 7 years for financial documents), and these durations are embedded as configuration parameters in data management systems. After implementation, these retention periods remain unchanged even when consent is revoked, processing purposes are fulfilled or narrowed, new legal requirements emerge, or interpretations of regulations evolve. This static approach overlooks the inherently dynamic nature of storage limitations under GDPR. The necessity for keeping a data element is not a static characteristic; it is a relational characteristic influenced by the interplay of (i) the processing purpose for which consent was obtained, (ii) the current status of that consent, (iii) the legal basis that legitimises the processing, and (iv) any conflicting legal obligations (e.g., anti-money laundering retention requirements, litigation holds) that could take precedence over the GDPR retention period. Whenever any of these factors changes, the obligation to retain must be re-evaluated.

From a language engineering standpoint, the challenge is clear: how can a metamodel-based modelling language formally delineate the dynamic interplay between consent state, processing purpose, legal basis, and retention obligation, ensuring that storage limitation constraints are (i) formally specified within the language definition, (ii) automatically validated during modelling, and (iii) traceable to specific regulatory provisions?

This paper fills this gap by proposing a metamodel extension for consent-derived dynamic storage limitation, intended to integrate with the ConsentML modelling language [12] and the DIAPROD framework [11], presented in section 2. The extension provides the following contributions:

A Retention Policy Model ($\mathcal{R}$) that formally binds retention periods to processing purposes and consent states, with temporal typing, jurisdictional overrides, and legal hold mechanisms (Section 4). **Operational semantics for purpose-linked erasure** specifying transition rules governing how data transitions through active, expired, held, and erased states in response to purpose completion, consent withdrawal, retention expiry, and legal hold events (Section 4). **A catalogue of well-formedness predicates** encoding GDPR storage limitation invariants (Art. 5(1)(e), Art. 17, Art. 25) as system-level constraints,

integrated into the Certifying Kernel’s constraint registry. **An illustrative case study** in the UK open banking domain demonstrating the expressiveness of the formal model across six storage limitation scenarios (Section 5).

Problem Statement. A fundamental disconnect persists between GDPR’s storage limitation mandate and its engineering realisation [13, 15]. The “necessity” of retaining a data element is not a fixed temporal property; it is a *relational property* determined by the intersection of (i) the processing purpose for which consent was granted, (ii) the current state of that consent, (iii) the legal basis under which the processing is legitimised, and (iv) any competing legal obligations that may override the default retention period. When any of these elements changes, the retention obligation must be reassessed. Yet existing approaches offer limited support for expressing, validating, and enforcing this dynamic relationship [5, 3, 16, 6].

Current practice forces one of two unsatisfactory choices. The first is to define retention periods as static configuration parameters in data management systems—a compliance officer assigns fixed durations (e.g., “7 years for financial records”), and these are encoded as database TTL (time-to-live) values or cron-scheduled deletion jobs, entirely disconnected from the consent model that authorises the processing. The second is to embed retention logic in application code, where it becomes opaque to auditors, untestable against regulatory requirements, and impossible to validate against the consent lifecycle. Neither approach can detect that a data element should be erased when consent is withdrawn, but another legal basis still justifies its retention. Neither can suspend erasure when a litigation hold is imposed, then reassess when the hold is lifted. Neither can resolve the conflict when a data element is subject to a 2-year marketing retention under Art. 6(1)(a) and a 5-year AML (anti-money laundering) retention under Art. 6(1)(c) simultaneously.

These limitations have practical consequences. The EDPB’s 2025 coordinated enforcement action [13] found that among 764 controllers audited across Europe, persistent weaknesses existed in systematic data classification and automated deletion capabilities—precisely the gap between static retention schedules and the dynamic, consent-derived enforcement that GDPR requires.

Research Questions. This paper addresses a specific open challenge in data privacy management: *how can a data protection framework enforce GDPR storage limitation at runtime through formally grounded, consent-derived mechanisms?* Existing approaches treat retention as a static configuration parameter; this work demonstrates that retention can instead be derived from the consent lifecycle state and enforced through language-level constraints. The contribution operates at three levels: the Retention Policy Model provides a formal specification of retention constraints bound to processing purposes and consent states; the erasure transition rules define how retention violations are detected and purpose-linked erasure is scheduled when consent is withdrawn, purposes complete, or retention periods expire; and the well-formedness predicate catalogue extends the Certifying Kernel of ConsentML [12] with eight storage limitation

invariants traceable to GDPR Articles 5(1)(e), 17, and 25. Together, these introduce five new metamodel classes, 21 relationships [10], and eight predicates into the DIAPROD (DynamIc dAtA PROtection by Design) framework [11], without modifying any existing class definitions.

Scope and Boundary. The mechanism presented in this paper operates at the *modelling and specification level*: it determines *when* data should be retained, held, archived, or erased based on the evolution of consent, processing purposes, legal bases, temporal conditions, and legal holds. It does not implement or verify the physical deletion of data from operational databases, replicas, backups, or trained machine learning models. The operational semantics (Section 4) define *state transitions* over retention states—formal objects in the metamodel—not commands to a storage engine. The term “enforcement” refers to the deterministic application of transition rules and the validation of well-formedness predicates at modelling time, ensuring that the specified retention policies are consistent, GDPR-compliant, and free of the violation classes identified in this paper (multi-purpose conflicts, undocumented holds, post-withdrawal retention). Translating these validated specifications into operational system behaviour—issuing deletion commands, verifying erasure across replicas, triggering machine unlearning procedures—is a complementary engineering concern addressed by runtime enforcement systems and is outside the scope of this work.

2 Background

The DIAPROD Framework. DIAPROD (DynamIc dAtA PROtection by Design) is a privacy engineering framework that reconceptualises Data Protection by Design (DPbD) as a runtime enforcement capability [11]. The framework is organised into an untrusted zone and a trusted zone separated by a formal trust boundary. The untrusted zone accepts natural-language consent specifications through an LLM-assisted layer that proposes candidate consent models. The trusted zone comprises a deterministic Certifying Kernel that validates all proposals against ConsentML’s [12] 14-tuple denotational semantics, seven lifecycle transitions, and six invariants (I_1 – I_6), and a tamper-evident persistent store. The kernel’s constraint registry is extensible: this paper contributes the storage limitation extension (C-SL-1..8), which registers alongside the existing consent predicates without modifying them. The full architecture diagram, machine-readable case study instance, and supplementary artefacts are publicly available on Zenodo [10].

The DIAPROT Modelling Tool. The DIAPROD framework is implemented in DIAPROT (DynamIc dAtA PROtection Tool), a standalone web-based modelling tool built for the practical implementation of the DIAPROD framework. DIAPROT provides a visual consent modelling canvas where practitioners create consent models by placing nodes (actors, data elements, purposes, legal bases, retention policies) and drawing typed relationships between them. The

tool enforces metamodel constraints during modelling—only valid relationships can be drawn between node types—and an integrated constraint validator executes all well-formedness predicates with a single button press, producing structured results showing the rule identifier, the affected element, the GDPR article, the severity, and a diagnostic message. The DIAPROT modelling environment: (a) consent modelling canvas with typed nodes and relationships, (b) property panel with dropdown enumerations for node attributes, and (c) constraint validator is deposited on Zenodo [10].

The ConsentML Foundation. ConsentML [12] is a metamodel-based visual modelling language for dynamic consent, developed as part of the DIAPROT [11] framework. The Consent construct is formally defined as a 14-tuple mapped to a product domain. Operational semantics define seven lifecycle transitions with preconditions, postconditions, and cascading effects. A Certifying Kernel provides deterministic validation against GDPR-grounded well-formedness predicates through an extensible constraint registry. The companion data minimisation extension [9] introduces the Purpose–Data Linkage Model, formalising which data is necessary for each purpose. The storage limitation extension presented here builds on both, connecting retention periods to purpose–data linkages and consent states.

Motivating Example. Consider a United Kingdom (UK) open banking ecosystem in which Zipfin Ltd, a data controller, processes consumer data through three data processors: ThetaRay (a sanction screening company), Volume (a payment processing company), and Nomad (an email marketing company). James Bond (a data subject) has granted two consents: c_{001} (payment and remittance, Art. 6(1)(b)) and c_{002} (email marketing, Art. 6(1)(a)). Under a static retention paradigm, the compliance officer assigns fixed periods: “6 years for financial records (UK Limitation Act 1980),” “2 years for marketing data,” “5 years for KYC (know your customer) data (UK Money Laundering Regulations (MLR) 2017).” These are encoded as database TTL values.

Now consider four dynamic scenarios that such a static approach cannot handle: *Scenario 1 (Consent Withdrawal)*: James Bond withdraws c_{002} . The marketing email must be erased under Art. 17(1)(b), but the same email may be retained under c_{001} for transaction notifications. A static TTL cannot distinguish these two purpose-dependent retention obligations. *Scenario 2 (Purpose Completion)*: ThetaRay, as part of an anti-fraud operation, performs a one-time identity verification on James Bond. Name and date of birth were retained for this purpose, but ThetaRay still requires them for ongoing sanction screening under c_{001} . The retention obligation shifts from one purpose to another—static schedules cannot model this transition. *Scenario 3 (Legal Hold)*: Zipfin receives a litigation hold notice. Data that would otherwise be erased upon purpose completion must be preserved until the hold is lifted, overriding the default retention period. This is among the first modelling languages to express the conflict between the obligation to erase and the legal hold. *Scenario 4 (Cross-Jurisdictional Conflict)*: UK AML regulations require 5-year KYC retention, while the mar-

keting consent permits only 2-year retention. For data elements subject to both obligations, the longer retention period prevails — but only for AML purposes. The marketing processing must still respect the 2-year limit. The contributions presented in this paper are set to address all six scenarios.

3 Related Work

This section examines existing approaches to data retention enforcement and identifies the gaps that motivate the storage limitation extension. Basin, Debois, and Hildebrandt [5] present the foundational formal framework for GDPR compliance, identifying “purpose” with “process” and demonstrating how to automatically generate privacy policies and detect violations of data minimisation at the modelling level. Their Data Processing Language (DPL) supports temporal constraints on data deletion, and data is deleted from the system when the deadline arrives. However, DPL treats retention periods as static policy parameters without consent-lifecycle integration or dynamic purpose-linked reassessment. Arfelt, Basin, and Debois [3] extend this work to monitor GDPR compliance using metric first-order temporal logic (MFOTL), demonstrating that temporal properties of data processing — including retention deadlines—can be formally monitored at runtime. Their MonPoly tool implements this monitoring. However, the monitoring operates over externally defined policies; it does not derive retention constraints from a formal consent model. Hublet, Basin, and Krstić [18] demonstrate real-time policy enforcement with metric first-order temporal logic in ESORICS 2022, showing that GDPR temporal constraints can be enforced in real time. Their approach enforces general temporal policies but does not connect to consent lifecycle semantics or purpose-dependent retention. The authors of [14, 7] provide formal frameworks for the right to be forgotten using cryptographic tools, addressing deletion-as-confidentiality without connecting to consent models or retention scheduling.

ConsentML [12] provides the formal consent modelling foundation upon which this extension builds. The existing ConsentML metamodel includes *TemporaryCondition* with `validFrom` and `validUntil` attributes, but does not formalise retention as a purpose-dependent, consent-derived constraint with erasure transitions and legal hold mechanisms. TILT [16] and its companion toolbox, TIRA [17], document retention periods in machine-readable transparency information, but without formal enforcement semantics. DPL [5] supports temporal deletion constraints but treats them as static policy parameters. RSL-IL4Privacy [6] enables privacy policy specification but does not model the dynamic relationship between consent state and retention obligation. DEFEND [20] provides a Privacy by Design (PbD) platform with runtime monitoring but without formal retention semantics or consent-derived erasure enforcement. Gharib et al. [15] identify challenges translating GDPR principles into engineering requirements, confirming the need for formal retention specification but without providing them.

State-machine representations of data retention policies are not new. GEO-DAC [2] uses a state-machine model in which each state represents a data life-

cycle stage (active, archived, deleted), enabling enforcement of retention policies in outsourced service environments. Barati et al. [4] formalise GDPR rules using timed automata and verify data flow compliance in business processes via the UPPAAL model checker. Both approaches demonstrate that retention can be modelled as state transitions with temporal constraints. However, neither approach derives retention decisions from the *consent lifecycle*. GEODAC’s state machine is triggered by fixed temporal events (a clock expires, an administrator intervenes); it has no concept of consent withdrawal contracting the permissible data scope, or of purpose completion triggering retention reassessment. Barati et al.’s timed automata verify whether a business process *conforms* to GDPR timing constraints but do not model the *consent state* that determines which constraints apply. Our contribution is not the state-based representation itself—it is the formal integration of consent evolution, processing purposes, legal bases, temporal conditions, and legal holds into a single metamodel extension where retention decisions are *derived from* the consent lifecycle state formalised in ConsentML, not imposed independently of it.

Table 1 summarises the coverage of existing approaches presented in this section across five dimensions necessary for consent-derived dynamic storage limitation.

To the best of our knowledge, no existing approach satisfies all five dimensions. The contribution of this paper is a formal metamodel extension that closes this gap by deriving storage limitation constraints directly from the consent lifecycle state formalised in ConsentML [12].

Table 1. Coverage of existing approaches across storage limitation dimensions. ✓ = Supported; P = Partial; ✗ = Not supported.

Approach	Formal Model	Purpose-Linked Retention	Consent Derived	Erasure Transitions	Regulatory Traceability
Basin et al. (DPL) [5]	✓	P	✗	P	P
Arfelt et al. (MonPoly) [3]	✓	✗	✗	P	P
Hublet et al. [18]	✓	✗	✗	P	P
Garg et al. [14]	✓	✗	✗	P	✗
Cohen et al. [7]	✓	✗	✗	P	✗
GEODAC [2]	P	P	✗	P	✗
Barati et al. [4]	✓	✗	✗	✗	P
TILT/TIRA [16, 17]	✗	P	✗	✗	P
DEFEND [20]	✗	P	✗	P	P
ConsentML [12]	✓	✗	P	✗	✓
This paper	✓	✓	✓	✓	✓

4 The Storage Limitation Extension

This section presents the storage limitation extension and introduces five new metamodel classes and 21 new relationship types [10] into the ConsentML metamodel [12]. The central construct is the *Retention Policy* ($\mathcal{R}$), which formally binds a retention period to a processing purpose and legal basis. The *Retention State* ($\mathcal{RS}$) represents the current retention status of a data element relative to a specific purpose. The *Erasure Trigger* ($\mathcal{ET}$) captures the event that initiates an erasure evaluation. The *Legal Hold* ($\mathcal{LH}$) represents a temporary override suspending erasure. The *Retention Audit Record* ($\mathcal{RA}$) logs all retention decisions for accountability.

The five classes were derived from GDPR provisions that existing ConsentML [12] constructs cannot express. **RetentionPolicy** (Art. 5(1)(e)) binds a retention duration to a specific processing purpose and legal basis—for instance, 2190 days for payment data under the UK Limitation Act versus 730 days for marketing data under consent—making the purpose-dependency explicit where static schedules treat retention as a global parameter. **RetentionState** (Art. 17(1)(a)) introduces a five-state lifecycle (active, expired, held, erased, archived) with computed expiry ($t_{exp} = t_{act} + \delta_{rp}$), enabling automated detection of elapsed retention. **ErasureTrigger** (Art. 17(1)(a)(b)) captures the event initiating erasure evaluation across five trigger types: retention expiry, consent withdrawal, purpose completion, erasure request, and legal obligation end. **LegalHold** (Art. 17(3)) suspends erasure without destroying the underlying policy. **RetentionAuditRecord** (Art. 5(2)) logs every retention decision; no trigger or hold may exist without a corresponding audit record. The storage-limitation metamodel extension is available in the supplementary materials [10].

Classes and Relationships. The syntax of the five classes is expressed in a BNF-style grammar. Backus–Naur Form (BNF) is a formal metalanguage used to define the syntax and grammar rules of computer languages [12].

Listing 1 defines the syntactic structure¹ of the **RetentionPolicy**. It consists of seven components: **Identifier**, which uniquely distinguishes the retention policy instance; **ProcessingPurpose**, which binds the policy to the specific purpose for which data is retained; **LegalBasis**, which identifies the GDPR legal ground (Art. 6(1)) legitimising the retention; **RetentionDuration**, a positive temporal value specifying the retention period in days; **RetentionJustification**, a natural-language statement explaining why the specified duration is necessary

¹ Due to space constraints, we intentionally omitted the syntactic structures of the other four classes. But essentially, they all follow a similar structural definition. The **RetentionState** comprises Identifier, PersonalData, RetentionPolicy, Status, ActivatedAt, ExpiresAt and CurrentState. The **ErasureTrigger** comprises Identifier, TriggerType, AffectedData, SourceEvent and Timestamp. The **LegalHold** comprises Identifier, HoldType, Justification, AffectedData, ImposedAt, LiftedAt and GDPRException. The **RetentionAuditRecord** comprises Identifier, Event, Decision, RetentionState, GDPRArticle and Timestamp.

(e.g., “UK Limitation Act 1980”); **JurisdictionalOverride**, an optional mapping from jurisdiction to alternative duration for cross-border scenarios where local regulations impose longer or shorter retention periods; and **ReviewPeriod**, a duration specifying how frequently the retention obligation must be reassessed to ensure continued necessity under Art. 5(1)(e).

```
RetentionPolicy ::= rp(Identifier, ProcessingPurpose, LegalBasis,
    RetentionDuration, RetentionJustification,
    JurisdictionalOverride, ReviewPeriod)
```

Listing 1: Syntax definition for the `RetentionPolicy` construct.

Retention Policy Model. We define the semantic mapping for the Retention Policy:

$$\llbracket rp(\text{Id}, PP, LB, Dur, Just, JO, Rev) \rrbracket = \langle id, pp, lb, \delta, just, jo, rev \rangle$$

where:

$$\langle id, pp, lb, \delta, just, jo, rev \rangle \in \mathcal{R} = I_R \times PP \times L_{BS} \times \Delta \times \mathcal{J} \times \mathcal{O} \times \Delta_{rev}$$

Δ denotes the temporal duration domain (a positive real number representing days). $\mathcal{J}$ denotes the set of natural-language justifications. $\mathcal{O}$ denotes the set of jurisdictional overrides (each mapping a jurisdiction to an alternative duration). Δ_{rev} denotes the review period duration.

Retention State Semantics. The retention state of a personal data element pd relative to a retention policy rp is:

$$\llbracket rs(\text{Id}, PD, RP, Stat, Act, Exp, CS) \rrbracket = \langle id, pd, rp, stat, t_{act}, t_{exp}, cs \rangle$$

where $t_{exp} = t_{act} + \delta_{rp}$ (the expiry timestamp is the activation timestamp plus the retention duration), and $cs \in \{active, expired, held, erased, archived\}$.

Retention Duration Derivation. The effective retention duration δ_{eff} for a data element pd under purpose pp is derived from the consent state, legal basis, and jurisdictional context:

$$\delta_{eff}(pd, pp) = \begin{cases} \delta_{rp} & \text{if } c.stat = granted \wedge jo = \emptyset \\ \delta_{jo}(j) & \text{if } c.stat = granted \wedge j \in jo \\ 0 & \text{if } c.stat \in \{withdrawn, objected\} \wedge \neg \exists lh \\ \delta_{lh} & \text{if } \exists lh : covers(lh, pd) \end{cases}$$

where $\delta_{jo}(j)$ is the jurisdictional override duration for jurisdiction j , δ_{lh} extends the retention until the legal hold is lifted, and the third case triggers erasure evaluation when consent is withdrawn, and no legal hold applies.

Multi-Purpose Retention Resolution. A single data element may be subject to multiple retention policies when it serves more than one processing purpose. For example, in the ZipFin case study presented in section 5, `email_address` serves both payment notification (pp_{001} , retention 2190 days under Art. 6(1)(b)) and marketing (pp_{002} , retention 730 days under Art. 6(1)(a)). When marketing consent c_{002} is withdrawn, the retention obligation for marketing expires, but the payment retention remains active. The effective retention for a data element is computed as the maximum across all active purpose-specific retentions:

$$\delta_{eff}(pd) = \max_{pp_i \in PP_{pd}^{active}} \delta_{eff}(pd, pp_i) \quad (1)$$

where PP_{pd}^{active} is the set of processing purposes for which pd (personal data) has an active retention state, and $\delta_{eff}(pd, pp_i)$ is the remaining retention duration under purpose pp_i . When $PP_{pd}^{active} = \emptyset$ —no active purpose justifies continued retention—the data element becomes an erasure candidate. The composition rule is deterministic and unambiguous: it always selects the longest remaining retention, ensuring that no data element is erased while any active purpose still requires it (satisfying Art. 5(1)(b)), while also ensuring that data is not retained beyond the last justifying purpose (satisfying Art. 5(1)(e)).

Operational Semantics for Purpose-Linked Erasure. We define five transition rules governing how retention states evolve. Each rule follows the same inference notation used for the ConsentML [12] lifecycle transitions.

Retention Expiry. When the retention period expires and no other purpose justifies continued storage:

$$\frac{t_{now} \geq t_{exp} \wedge \neg \exists pp' \neq pp : active(rp', pd) \wedge \neg \exists lh : covers(lh, pd)}{\langle rs, \sigma \rangle \xrightarrow{\text{expiry}} \langle rs[cs:=erased], \sigma' \cup \{\mathcal{ET}_{expiry}, \mathcal{RA}_{expiry}\} \rangle} [\text{RetentionExpiry}]$$

Consent Withdrawal Erasure. When consent is withdrawn under Art. 7(3), the retention obligation for all data held solely under that consent must be reassessed:

$$\frac{c.stat=withdrawn \wedge \neg \exists lb' : lb' \neq consent \wedge legitimises(lb', pa, pd)}{\langle rs, \sigma \rangle \xrightarrow{\text{withdraw-erase}} \langle rs[cs:=expired], \sigma' \cup \{\mathcal{ET}_{withdrawal}\} \rangle} [\text{WithdrawalErasure}]$$

This rule links directly to ConsentML’s WITHDRAW transition. If another legal basis (e.g., legitimate interest, legal obligation) justifies continued processing, the data is not erased—satisfying Art. 17(1)(b)’s condition “and there is no other legal ground for the processing.”

Purpose Completion Erasure. When a processing purpose completes and no other active purpose requires the data:

$$\frac{complete(pp) \wedge \neg \exists pp' \neq pp : requires(pp', pd) \wedge active(pp')}{\langle rs, \sigma \rangle \xrightarrow{\text{complete-erase}} \langle rs[cs:=expired], \sigma' \cup \{\mathcal{ET}_{completion}\} \rangle} [\text{CompletionErasure}]$$

ID (Article)	Well-Formedness Predicate
C-SL-1 (Art. 25(2))	$\forall pp \in PP : \exists rp \in \mathcal{R} : bindsTo(rp, pp)$
C-SL-2 (Art. 5(1)(e))	$\forall rp \in \mathcal{R} : defined(rp.\delta) \wedge rp.\delta > 0 \wedge defined(rp.just)$
C-SL-3 (Art. 17(1)(a))	$\forall rs \in \mathcal{RS} : t_{now} \geq rs.t_{exp} \Rightarrow rs.cs \in \{expired, held, erased, archived\}$
C-SL-4 (Art. 17(1)(b))	$\forall pd \in PD, c \in C : c.stat = withdrawn \wedge basedOn(pd, c) \Rightarrow reassessed(rs_{pd})$
C-SL-5 (Art. 17(3))	$\forall lh \in \mathcal{LH} : defined(lh.exception) \wedge lh.exception \in \{17(3)(b), 17(3)(d), 17(3)(e)\}$
C-SL-6 (Art. 5(1)(b))	$\forall pd \in PD : PP_{pd}^{active} > 1 \Rightarrow \delta_{eff}(pd) = \max_{pp_i} \delta_{eff}(pd, pp_i)$
C-SL-7 (Art. 17(3))	$\forall rs \in \mathcal{RS} : rs.cs = held \Rightarrow \neg \exists et : erases(et, rs.pd)$
C-SL-8 (Art. 5(2))	$\forall et \in \mathcal{ET} : \exists ra \in \mathcal{RA} : records(ra, et) \wedge defined(ra.article)$

Table 2. Well-formedness predicates with governing GDPR articles.

This rule connects storage limitation to data minimisation: the data minimisation extension’s PURPOSECOMPLETIONCONTRACTION rule contracts the permissible data scope, and this rule triggers the corresponding retention expiry.

Legal Hold Imposition. When a legal hold is imposed, the affected data transitions to the *held* state, suspending any pending erasure:

$$\frac{impose(lh) \wedge covers(lh, pd) \wedge rs.cs \in \{active, expired\}}{\langle rs, \sigma \rangle \xrightarrow{hold} \langle rs[cs:=held], \sigma' \cup \{\mathcal{RA}_{hold}\} \rangle} \text{ [HoldImposition]}$$

The hold does not destroy the underlying retention policy; it suspends erasure. When the hold is lifted, the retention state reverts to its previous state for reassessment.

Legal Hold Release. When a legal hold is lifted, the previously held data is reassessed:

$$\frac{lift(lh) \wedge covers(lh, pd) \wedge rs.cs = held}{\langle rs, \sigma \rangle \xrightarrow{release} \langle rs[cs:=s'], \sigma' \cup \{\mathcal{RA}_{release}\} \rangle} \text{ [HoldRelease]}$$

where $s' = active$ if $t_{now} < t_{exp}$, and $s' = expired$ if $t_{now} \geq t_{exp}$.

5 Illustrative Case Study

This section applies the storage limitation extension to an illustrative scenario in the United Kingdom’s open banking ecosystem, demonstrating how the mechanism detects violation classes that static retention schedules miss. The complete

model was constructed in the **DIAPROT** modelling tool presented in Section 2. The resulting model of the case study is presented in the supplementary material [10]. *Model Structure:* In this case study, ZipFin Ltd (data controller) delegates processing to three Data Processors — ThetaRay (sanction screening company), Volume (payment processing company), and Nomad (email marketing company). James Bond (data subject) grants two consents: c_{001} (payment and remittance, supported by Contract Art. 6(1)(b)) and c_{002} (email marketing, supported by Consent Art. 6(1)(a), subsequently withdrawn). Seven Personal Data elements are modelled as separate nodes (**acct_no**, **sort_code**, **tx_ref**, **nationality**, **name**, **dob**, **email**), each connected to James Bond via **owns** edges. Critically, **email** serves two purposes: transaction notifications under c_{001} and promotional communications under c_{002} —creating the multi-purpose retention conflict central to Scenario S2. Three Processing Purposes (pp_{001} – pp_{003}) are defined by Zipfin Ltd via **defines** edges and linked to their respective consents via **for** edges. The storage limitation extension layer overlays this base with retention-enforcement constructs connected through 21 typed relationships, presented in the supplementary material [10]. *Retention Policy Definitions:* Three Retention-Policy instances are created, one per processing purpose, and connected to the base model via **hasRetention** (from ProcessingPurpose) and **basedOn** (to Legal-Basis):

Payment Services (rp_{001}): 6-year retention for financial records and transaction notification history.

$$rp_{001} = \langle pp_{001}, \text{Art.6(1)(b)}, \underbrace{2190}_{6 \text{ years}}, \text{UK Limitation Act}, \emptyset, \underbrace{365}_{1 \text{ year}} \rangle$$

As shown in the model file "**storage_limitationcase_study_model.png**", in the supplementary materials [10], Payment Services connects to **RP_Payment_6yr** via **hasRetention**, and **RP_Payment_6yr** connects to Contract Art. 6(1)(b) via **basedOn**. Four **appliesTo** edges link the policy to **acct_no**, **sort_code**, **tx_ref**, and **email**—the last because payment transaction notifications require the data subject's email address.

Sanction Screening (rp_{002}): 5-year KYC retention with EU jurisdictional override.

$$rp_{002} = \langle pp_{002}, \text{Art.6(1)(c)}, \underbrace{1825}_{5 \text{ years}}, \text{UK MLR 2017}, \{EU \mapsto 3650\}, \underbrace{365}_{1 \text{ year}} \rangle$$

RP_Screening_5yr connects to Legal Obligation Art. 6(1)(c) via **basedOn**, with **appliesTo** edges to **name**, **dob**, and **nationality**. The jurisdictional override ($jo_{EU} = 3650$ days) is encoded as a policy attribute, activated when the processing jurisdiction is EU.

Email Marketing (rp_{003}): 2-year consent-based retention.

$$rp_{003} = \langle pp_{003}, \text{Art.6(1)(a)}, \underbrace{730}_{2 \text{ years}}, \text{Marketing best practice}, \emptyset, \underbrace{180}_{6 \text{ months}} \rangle$$

RP_Marketing_2yr connects to Consent Art. 6(1)(a) via **basedOn**, with an **appliesTo** edge to **email**. Note that **email** is now subject to *two* retention

policies (rp_{001} and rp_{003}) under different legal bases—a static TTL value cannot distinguish these two purpose-dependent obligations. C-SL-1 and C-SL-2 defined in Table 2 are satisfied for all three purposes.

Retention State Instantiation. Three `RetentionState` instances demonstrate the multi-purpose retention model, each connected through three edges: `tracksRetention` (to `RetentionPolicy`), `tracksData` (to `PersonalData`), and `derivedFromConsent` (to `Consent`). **RS_acct_no_active:** tracks `acct_no` under rp_{001} , derived from c_{001} (granted). Activated 2024-01-15, expires 2030-01-15. Current state: *active*. The `derivedFromConsent` edge to c_{001} confirms that the consent justifying retention remains granted—C-SL-4 is satisfied. **RS_email_payment_active:** tracks `email` under rp_{001} , derived from c_{001} (granted). Activated 2024-01-15, expires 2030-01-15. Current state: *active*. This retention state captures the payment purpose’s independent retention obligation over `email` for transaction notification records. It remains active regardless of what happens to c_{002} .

RS_email_marketing_expired: tracks `email` under rp_{003} , derived from c_{002} (withdrawn). Current state: *expired*. The `derivedFromConsent` edge to c_{002} reveals that consent has been withdrawn. Because the state is *expired* (not *active*), C-SL-4 is satisfied. C-SL-6 fires as a warning for `email` because it has two `RetentionStates`—the validator reminds the modeller to verify that the longest period (6-year payment retention) prevails, which it does: $\delta_{eff}(\text{email}) = \max(0, 2190) = 2190$ days.

Scenario Evaluation. The six scenarios below trace paths through the model presented in the supplementary materials [10], demonstrating how the transition rules and well-formedness predicates presented in Table 2 interact with the graph structure.

S1: Retention Expiry (C-SL-3). The 2-year marketing retention for rp_{003} expires. The `RETENTIONEXPIRY` rule traverses the `tracksRetention` edge from `RS_email_marketing_expired` to `RP_Marketing_2yr`, computes $t_{exp} = t_{act} + 730$, and confirms $t_{now} \geq t_{exp}$. It then follows the `tracksData` edge to `email` and checks whether any other `RetentionState` for `email` remains active; thus, `RS_email_payment_active` is still active under c_{001} . Therefore, `email` is *not* erased: `RS_email_marketing_expired` transitions to *erased* for the marketing purpose, but `email` itself survives under the payment retention until 2030. A `RetentionAuditRecord` is generated citing Art. 17(1)(a). C-SL-3 validates that no expired data remains in the *active* state.

S2: Consent Withdrawal Erasure (C-SL-4, C-SL-6). James Bond withdraws c_{002} before the 2-year period expires. The `WITHDRAWALERASURE` rule follows the `derivedFromConsent` edge from `RS_email_marketing_expired` to c_{002} , detects the *withdrawn* state, and evaluates whether an alternative retention obligation justifies continued storage. The rule traverses the `tracksData` edge to `email` and discovers `RS_email_payment_active`—still active under c_{001} with rp_{001} (6-year payment retention). C-SL-6 computes $\delta_{eff}(\text{email}) = \max(0, 2190) = 2190$ days: the payment retention prevails. `RS_email_marketing_expired` transitions to *expired*, and the ErasureTrigger `ET_Marketing-Withdrawal` is emit-

ted with a `triggeredByConsent` edge to c_{002} and a `producesCandidate` edge to `EC_email`. The `EraseCandidate` records `email` with status *retained_other_purpose* and remaining purpose pp_{001} —confirming that the data subject’s email is retained solely for payment transaction notifications, not for marketing. `RAudit_Withdrawal` records the event via `records` and cites Consent Art. 6(1)(a) via `citesArticle`. This scenario demonstrates the core contribution: a static TTL cannot distinguish these two purpose-dependent retention obligations, and would either erase `email` (breaking payment notifications) or retain it indefinitely (violating the marketing withdrawal).

S3: Legal Hold Suspension (C-SL-5, C-SL-7). Zipfin receives a litigation hold from the ICO covering all financial data. As shown in the model, `LH_Litigation_Jones` connects to `RS_acct_no_active` via `suspends`, to `acct_no`, `sort_code`, and `tx_ref` via `affects`, to ICO via `imposedBy`, and to Contract Art. 6(1)(b) via `citesException` (citing Art. 17(3)(e), defence of legal claims). The `HOLDIMPOSITION` rule transitions `RS_acct_no_active` to *held*. C-SL-5 validates that the hold cites a valid Art. 17(3) exception. C-SL-7 enforces the critical invariant: `ET_Marketing_Withdrawal`’s `targets` edge points to

`RS_email_marketing_expired` (not to `RS_acct_no_active`), so no erasure trigger targets held data. `RAudit_LegalHold` records the imposition via `recordsHold`. When the hold is lifted, `HOLDRELEASE` reassesses: if $t_{now} < t_{exp}$, data reverts to *active*; otherwise, it transitions to *expired*.

S4: Purpose Completion (C-SL-3, linking to data minimisation). Suppose the identity verification purpose completes. The `COMPLETIONERASURE` rule checks whether `name` and `dob` are required by any other active purpose by traversing the `tracksData` edges from all `RetentionStates` for these elements. Since `ThetaRay` requires them for ongoing sanction screening under rp_{002} (connected via `appliesTo`), they are not erased—the retention obligation shifts from the completed purpose to the active one. This demonstrates the formal link between the storage limitation extension and the data minimisation extension’s `PURPOSECOMPLETIONCONTRACTION` rule: data is retained only as long as at least one active purpose justifies it.

S5: Cross-Jurisdictional Override. Zipfin operates under both UK MLR (5-year KYC retention) and EU AMLD (5-year retention plus 5-year archive). For KYC data processed in an EU jurisdiction, rp_{002} includes a jurisdictional override: $jo_{EU} = 3650$ days. C-SL-6 computes $\delta_{eff}(\text{nationality}, pp_{002}) = 3650$ for the EU context while maintaining $\delta_{eff}(\text{nationality}, pp_{002}) = 1825$ for the UK context—ensuring that jurisdictional requirements are satisfied without over-retaining data in jurisdictions with shorter periods.

S6: Audit Trail Completeness (C-SL-8). Every event in S1–S5 produces a `RetentionAuditRecord`. In the model, `RAudit_Withdrawal` connects to `RS_email_marketing_expired` via `audits`, to `ET_Marketing_Withdrawal` via `records`, and to Consent Art. 6(1)(a) via `citesArticle`. `RAudit_LegalHold` connects to `RS_acct_no_active` via `audits`, to `LH_Litigation_Jones` via `recordsHold`, and to Contract Art. 6(1)(b) via `citesArticle`. C-SL-8 validates that no `EraseTrigger` or `LegalHold` exists without a corresponding audit record with a

defined GDPR article citation. The kernel rejects any model where this invariant is violated.

6 Discussion

Contribution to Data Privacy Management. The core contribution is the demonstration that storage limitation—typically treated as a static retention schedule—can be formalised as a *consent-derived, purpose-dependent enforcement mechanism*. The five-state retention lifecycle (active, expired, held, erased, archived) replaces the binary “retained/deleted” distinction implicit in current practice with a formal vocabulary that captures intermediate states invisible to static approaches: data whose retention has expired but whose erasure is suspended by a legal hold, or data whose marketing retention has lapsed but whose payment retention remains active. Scenario S2 demonstrates a violation class which makes this paper among the first approaches to detect: **email** subject to two purpose-dependent retention obligations under different legal bases, where withdrawal of one consent must not trigger erasure justified by the other. **Integration with DIAPROD.** The extension connects to the DIAPROD framework at four points: RetentionPolicy links to ConsentML’s ProcessingPurpose and LegalBasis; the WITHDRAWALERASURE rule extends ConsentML’s WITHDRAW transition with retention reassessment; the COMPLETIONERASURE rule bridges to the data minimisation extension’s PURPOSECOMPLETIONCONTRACTION via shared ErasureCandidate constructs; and C-SL-1 through C-SL-8 use the same Certifying Kernel constraint registry as existing predicates—requiring zero modifications to the base language. **Specification vs. Execution.** The mechanism operates at the specification level: it answers *what should happen* (which data should be erased, held, or retained) and *why* (which GDPR article, which consent state, which purpose). It does not answer *how* the erasure is executed across databases, replicas, and backups. This is a deliberate design choice: the specification layer must be correct and auditable independently of the execution layer. A correct specification executed incorrectly is a system implementation failure; an incorrect specification executed perfectly is a compliance failure. Our mechanism addresses the latter. The bridge between specification and execution is the subject of ongoing work within the DIAPROD framework’s runtime enforcement layer.

Limitations. The classification of retention periods requires domain expertise that the framework does not automate. The jurisdictional override mechanism supports single-level overrides but not multi-layered regulatory hierarchies (e.g., EU regulation overriding member state law). Legal holds are modelled as binary (imposed/lifted) without partial holds affecting specific data subsets within a held collection. The mechanism validates the retention policy structure at the model level, independent of the number of data subjects; per-subject retention evaluation is event-driven with $O(k)$ cost per consent event.

7 Conclusion

This paper presented a consent-derived retention enforcement mechanism introducing five metamodel classes, 21 typed relationships, five erasure transition rules, legal hold mechanisms, and eight GDPR-grounded compliance constraints. The mechanism demonstrates that storage limitation can be enforced as a dynamic, purpose-dependent property derived from the consent lifecycle—detecting violation classes that static retention schedules miss, including multi-purpose retention conflicts and erasure obligations blocked by undocumented legal holds. The mechanism is implemented in the DIAPROT modelling tool and validated through six scenarios in the ZipFin open banking case study. Future work includes summative evaluation across healthcare, financial services, and public-sector domains, and LLM-assisted retention-period recommendation with kernel-validated outputs.

References

1. Regulation (eu) 2016/679: General data protection regulation (gdpr). Official Journal of the European Union pp. 1–88 (2016), <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:02016R0679>
2. Agrawal, D., Chawla, S., Johnson, R., Stolfo, S.: GEODAC: A data assurance policy specification and enforcement framework for outsourced services. In: Proceedings of the 5th International Workshop on Data Management in Peer-to-Peer Systems. ACM (2009)
3. Arfelt, E., Basin, D., Debois, S.: Monitoring the GDPR, pp. 681–699 (2019). https://doi.org/10.1007/978-3-030-29959-0_33
4. Barati, M., Rana, O.F., Petri, I., Theodorakopoulos, G.: GDPR compliance verification in Internet of Things. *IEEE Access* **8**, 119697–119709 (2020). <https://doi.org/10.1109/ACCESS.2020.3005509>
5. Basin, D., Debois, S., Hildebrandt, T.: On Purpose and by Necessity: Compliance Under the GDPR, pp. 20–37 (2018). https://doi.org/10.1007/978-3-662-58387-6_2
6. Caramujo, J., da Silva, A.R., Monfared, S.N., Ribeiro, A., Calado, P., Breauux, T.D.: Rsl-il4privacy: a domain-specific language for the rigorous specification of privacy policies. *Requirements Engineering* **24**, 1 – 26 (2018), <https://api.semanticscholar.org/CorpusID:254087630>
7. Cohen, A., Smith, A., Swanberg, M., Vasudevan, P.N.: Control, confidentiality, and the right to be forgotten. In: Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. pp. 3358–3372. ACM (11 2023). <https://doi.org/10.1145/3576915.3616585>
8. Cormack, A.: An introduction to the gdpr (v2). IDPro Body of Knowledge **1** (6 2021). <https://doi.org/10.55621/idpro.11>
9. Daniels, S.O., Mouratidis, H.: Consent-derived data minimisation: A metamodel extension for dynamic purpose–data scope enforcement (2026), manuscript in preparation
10. Daniels, S.O., Mouratidis, H.: Supplementary materials for "from static schedules to consent-driven retention: An approach to gdpr storage limitation enforcement" (2026). <https://doi.org/10.5281/zenodo.21722452>

11. Daniels, S.O.: DIAPROD: A framework for formal consent modelling, dynamic storage limitation, data minimisation, and LLM-assisted compliance for data protection by design. In: Bernasconi, A., Cortès-Cornax, M., de la Vara, J.L., Nègre, E., Polacsek, T., Ralyté, J., Ruiz, M. (eds.) *Research Challenges in Information Science: RCIS 2026 Forum and Doctoral Consortium. Lecture Notes in Business Information Processing*, vol. 597, pp. 235–246. Springer, Cham (2027). https://doi.org/10.1007/978-3-032-35043-5_20
12. Daniels, S.O., Mouratidis, H.: ConsentML: A modelling language for dynamic consent modelling. In: Polacsek, T., Ruiz, M., Ralyté, J., Ravat, F. (eds.) *Research Challenges in Information Science. Lecture Notes in Business Information Processing*, vol. 585, pp. 542–557. Springer, Cham (2026). https://doi.org/10.1007/978-3-032-26836-5_33
13. European Data Protection Board: 2025 coordinated enforcement action: Implementation of the right to erasure by controllers. Tech. rep., EDPB (Feb 2026), adopted 10 February 2026. https://www.edpb.europa.eu/system/files/2026-02/edpb_cef-report_2025_right-to-erasure_en.pdf
14. Garg, S., Goldwasser, S., Vasudevan, P.N.: Formalizing Data Deletion in the Context of the Right to Be Forgotten, pp. 373–402 (2020). https://doi.org/10.1007/978-3-030-45724-2_13
15. Gharib, M., Salnitri, M., Paja, E., Giorgini, P., Mouratidis, H., Pavlidis, M., Ruiz, J.F., Fernandez, S., Siria, A.D.: Privacy requirements: Findings and lessons learned in developing a privacy platform. In: 2016 IEEE 24th International Requirements Engineering Conference (RE). pp. 256–265. IEEE (9 2016). <https://doi.org/10.1109/RE.2016.13>
16. Grünewald, E., Pallas, F.: TILT: A GDPR-aligned transparency information language and toolkit for practical privacy engineering. In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency (FAccT)*. pp. 636–646. ACM (2021). <https://doi.org/10.1145/3442188.3445925>
17. Grünewald, E., Wille, P., Pallas, F., Borges, M.C., Ulbricht, M.R.: Tira: An openapi extension and toolbox for gdpr transparency in restful architectures (6 2021). <https://doi.org/10.1109/EuroSPW54576.2021.00039>, <http://arxiv.org/abs/2106.06001><http://dx.doi.org/10.1109/EuroSPW54576.2021.00039>
18. Hublet, F., Basin, D., Krstić, S.: Real-Time Policy Enforcement with Metric First-Order Temporal Logic, pp. 211–232 (2022). https://doi.org/10.1007/978-3-031-17146-8_11
19. Orlando, A., Santoro, M.: A semantic approach to understanding gdpr fines: From text to compliance insights. *Computer Law & Security Review* (2025), <https://api.semanticscholar.org/CorpusID:281692192>
20. Piras, L., Al-Obeidallah, M.G., Mouratidis, H., Pavlidis, M.: DEFEND architecture: A privacy by design platform for GDPR compliance. In: *Trust, Privacy and Security in Digital Business (TrustBus 2019). LNCS*, vol. 11711, pp. 78–93. Springer (2019). https://doi.org/10.1007/978-3-030-27813-7_6
21. Shah, A., Banakar, V., Shastri, S., Wasserman, M., Chidambaram, V.: Analyzing the impact of gdpr on storage systems. *ArXiv abs/1903.04880* (2019), <https://api.semanticscholar.org/CorpusID:75134900>

A Privacy-Preserving Architecture for Secure Appliance-Level Demand Response in Virtual Power Plants

Joan Ferré-Queralt¹[0009-0002-0014-0065], Jordi Castellà-Roca¹[0000-0002-0037-9888], Alexandre Viejo¹[0000-0003-2342-5100], Nicolás Orduz Álvarez²[0009-0006-3776-7987], and Takao Tsuji²

¹ Universitat Rovira i Virgili

Departament d'Enginyeria Informàtica i Matemàtiques
CYBERCAT-Center for Cybersecurity Research of Catalonia
Av. Països Catalans 26, E-43007 Tarragona, Catalonia, Spain
{joan.ferreq,jordi.castella,alexandre.viejo}@urv.cat

² Yokohama National University

Graduate School of Engineering Science
Yokohama, Japan

nicolas-alvarez-zn@ynu.jp,tsuji-takao-pz@ynu.ac.jp

Abstract. Virtual Power Plants (VPPs) enable the aggregation of small flexible loads, allowing residential appliances to contribute to demand response (DR) programs. However, this participation requires fine-grained information about appliance availability, operational constraints, and user preferences, and may also involve remote appliance activation. As a result, appliance-level DR introduces privacy and security risks that are not sufficiently addressed by conventional aggregation approaches. In particular, availability reports, activation decisions, and delivery evidence may reveal sensitive behavioral patterns or enable unauthorized manipulation of DR operations. This paper proposes a privacy-preserving and secure architecture for appliance-level DR in VPPs. The framework combines appliance clustering, OpenADR-based activation, public-key cryptography, operational pseudonyms, and smart contracts to support verifiable DR coordination while reducing the exposure of user-specific information. The proposed design protects appliance availability from public disclosure, prevents unauthorized observers from linking DR actions to real users, and enables authenticated activation and evidence verification. A qualitative evaluation shows that the architecture satisfies the defined functional, privacy, and security requirements, providing a practical basis for secure and privacy-aware appliance participation in VPP-based DR programs.

Keywords: Demand response · Virtual Power Plant · Privacy · Security · Smart contracts

1 Introduction

Electric power systems must continuously maintain the balance between generation and demand. Imbalances may compromise system stability and lead to blackouts [7]. Traditionally, demand has had peaks and valleys that do not always coincide with the forecasting and this challenge is now becoming more acute. Electricity consumption continues to increase, placing additional stress on grid operation, while renewable energy resources have reached unprecedented levels of penetration [21]. These zero-emission sources are less predictable, which introduces less controllable fluctuations in power supply [11].

As a result, grid stability increasingly depends on robust ancillary services and advanced demand response (DR) mechanisms capable of preserving reliability.

Historically, DR programs focused mainly on large industrial and commercial consumers with significant load-reduction capacity. Early schemes were commonly based on interruptibility contracts or incentive-based mechanisms, in which participants agreed to reduce consumption when requested by the system operator [23]. This model offered system operators relatively large and reliable reductions from a limited number of participants. However, such programs provide limited granularity. Since flexibility is obtained from a small number of large consumers, the operator can only control demand at an aggregated and relatively coarse level.

To overcome these limitations, research has increasingly considered the use of smaller distributed loads as providers of ancillary services [6]. Some examples are residential appliances and other user-side resources, which have limited individual capacity, but their aggregate flexibility can become substantial when coordinated at scale.

Individual appliances generally cannot participate directly in service markets, where minimum capacity thresholds are required. In response, Virtual Power Plants (VPPs) have emerged as aggregators that combine residential and commercial resources into a larger virtual resource capable of satisfying market participation requirements [13]. VPPs must solve scheduling problems over heterogeneous loads by classifying them into similar (in terms of power and constraints) clusters and treating each cluster as one load.

Moreover, DR participation requires the exchange of fine-grained flexibility information and may involve remote control over user appliances. Flexibility data can reveal sensitive behavioral patterns of users, such as occupancy, daily routines, appliance usage and user preferences. At the same time, remote activation introduces security risks, since forged commands, manipulated schedules, or fraudulent participation evidence may affect user comfort, remuneration, and even grid operation. Consequently, security and privacy must be considered fundamental design requirements in any VPP-based demand response framework.

1.1 Related Work

Authors in [10, 14, 16, 18] proposed optimal demand aggregation frameworks in which VPPs schedule clustered flexible loads for market participation. However, these approaches provide only limited treatment of user-level comfort constraints. Other works [1, 4, 5, 17, 20] address such constraints while introducing different VPP load scheduling algorithms.

Nevertheless, these works generally assume that the VPP already has access to reliable information about the flexibility, availability, and operational constraints of participating loads. Before any DR activation can be scheduled, this information must first be collected. Since appliance-level availability and constraints may reveal sensitive information about users' habits, privacy must be considered from the design stage.

Privacy in DR systems involving user appliances as deferrable loads has received limited attention. Some works rely on cryptographic techniques, such as homomorphic encryption or Shamir's Secret Sharing, to allow appliance-related data to be shared with aggregators while limiting the disclosure of individual user information [8, 15, 22]. Other approaches [3, 9, 19] use consortium blockchain-based frameworks to preserve user privacy and maintain an immutable record of system operations.

However, these contributions typically address isolated parts of the DR workflow, such as privacy-preserving data aggregation or auditable transaction recording, rather than the complete appliance-level DR process. In particular, they do not jointly consider appliance setup and registration, the collection of fine-grained availability and operational constraints, the aggregation of appliances into schedulable flexibility clusters, the secure activation of end-user loads by a VPP, and the generation of verifiable evidence for remuneration. Moreover, when appliances are directly controllable by a third party, additional security issues arise, including activation authenticity, unauthorized control, evidence forgery, and the linkability of activation or delivery records to real users. These aspects remain insufficiently addressed in existing privacy-preserving DR and blockchain-based VPP proposals.

1.2 Contributions

This paper addresses the gap in the literature by proposing a privacy-preserving DR architecture for interoperable appliance-level aggregation, scheduling, activation, verification, and remuneration under VPP coordination. The main contributions of this work are:

- Propose a DR architecture for VPP-based appliance-level flexibility management, where residential and commercial appliances are registered, clustered, scheduled, activated, and remunerated through a unified framework.
- Introduce a privacy-preserving availability reporting mechanism, allowing the VPP to compute cluster-level flexibility without public disclosure of individual availability patterns.

- Define a secure activation in which appliances accept only authenticated events from the VPP.
- Design an evidence-verification procedure in which compensation is released only for signed evidence while preserving the privacy of activated appliances.

2 System Overview

The overall architecture of the proposed system is presented in this section. It introduces the main actors and components, summarizes their responsibilities, and defines the appliance-level parameters used throughout the protocol.

2.1 Actors and Components

The proposed framework involves several actors and technical components, each playing a specific role in the aggregation, scheduling, activation, and verification of demand response (DR) services.

Users A user U_i denotes an individual or entity enrolled in a DR program through a contractual agreement with a VPP. Each user owns a set of controllable appliances that can provide flexible power curtailment by adapting their operation in response to external DR signals. Through a mobile application, the user registers appliances, configures their operational preferences, and defines availability and comfort-related constraints.

Appliances Each appliance $a_{i,j}$ is owned by user U_i and represents a controllable load that can participate in the DR program. Appliances are characterized by a set of technical and operational parameters that define their curtailment capacity $P_{i,j}$ and user-defined constraints $\theta_{i,j} = (max_{i,j}, rec_{i,j})$.

Where $P_{i,j}$ denotes the nominal power of appliance $a_{i,j}$, $max_{i,j}$ denotes the maximum time that appliance $a_{i,j}$ must remain in OFF state, and $rec_{i,j}$ denotes the recovery time required after a curtailment event before appliance $a_{i,j}$ can participate in another DR activation.

We assume that each participating appliance contains a trusted control and measurement component connected to a hardware secure module (HSM). The HSM releases an operational signature for delivery evidence only if the appliance control logic verifies that the requested curtailment state was maintained during the activation interval.

Manufacturers Each manufacturer M_m is responsible for producing and provisioning appliances that can participate in the proposed DR framework. Manufacturers are trusted to equip appliances with secure hardware components. Manufacturers also provide certified information about appliance capabilities, including the nominal power value used during registration and clustering.

Clusters To perform the scheduling [12], the VPP groups appliances with similar curtailment power and operational constraints. Each cluster c is represented by $\theta_c = (P_c, \max_c, \text{rec}_c)$, where P_c is a guaranteed per-appliance curtailable power, rather than the total cluster capacity. The parameters are conservatively selected so that every appliance assigned to the cluster satisfies

$$P_{i,j} \geq P_c, \quad \max_{i,j} \geq \max_c, \quad \text{rec}_{i,j} \leq \text{rec}_c.$$

Equivalently, when computed from a fixed set of members, these values can be selected as the minimum curtailable power, minimum maximum-OFF time, and maximum recovery time, respectively. Hence, $N_{c,d}[t]P_c$ is a conservative estimate of the available cluster capacity.

The VPP defines the admissible ranges of these parameters for each cluster in a step previous to the setup. During registration, the RA compares $(P_{i,j}, \theta_{i,j})$ with these predefined cluster profiles and assigns $a_{i,j}$ to a compatible cluster through $C_c \leftarrow \text{AssignCluster}(P_{i,j}, \theta_{i,j})$. The particular partitioning of the parameter space is deployment-specific; the proposed security and privacy mechanisms are independent of the clustering algorithm used.

Cluster Smart Contract ($\mathcal{CSC}_c$) For each cluster c , the VPP deploys a cluster smart contract $\mathcal{CSC}_c$, which stores the cluster parameters and supports the later availability reporting, scheduling, activation, evidence verification, and payment procedures.

The contract therefore mainly performs authorization, state lookup, signature verification, replay checks, and payment accounting. Using key-value mappings, these checks have constant storage-access complexity per transaction. However, the total on-chain storage and transaction volume grow with the number of participating appliances and scheduled addresses. In large deployments, the scheduled set $\alpha_{c,d,I}$ could instead be represented by a Merkle-root commitment, with appliances providing membership proofs during evidence submission, reducing the on-chain schedule representation from $O(|\alpha_{c,d,I}|)$ to $O(1)$.

Virtual Power Plant The Virtual Power Plant (VPP) aggregates flexibility from clusters. This enables the VPP to treat a set of appliances as if it were only one load. The VPP also collects availability information, performs scheduling and issues DR activation commands. In the proposed architecture, the VPP acts as the main coordinator of the DR program and interfaces with electricity markets as an aggregated flexibility provider.

Registration Authority The Registration Authority (RA) verifies appliance registration information, assigns appliances to the appropriate flexibility clusters, and authorizes them to report curtailment availability.

2.2 Phases

The protocol is organized into three main phases: the *setup phase*, the *registration phase*, and the recurring *operational phase*.

Setup phase. During setup, the VPP, manufacturers, and system authorities generate and distribute the cryptographic material required by the protocol. The VPP defines the set of flexibility clusters and deploys the corresponding cluster smart contracts $\mathcal{CSC}_c$.

Registration phase. During appliance registration, each user U_i authenticates with the Registration Authority (RA) and enrolls their appliances $a_{i,j}$ in the DR program. For each appliance, the user provides the required operational preferences and constraints. Based on the appliance profile, the RA assigns the appliance to a flexibility cluster c .

Operational phase. After registration, the operational phase is executed periodically and consists of the following steps, which are also shown in Figure 1:

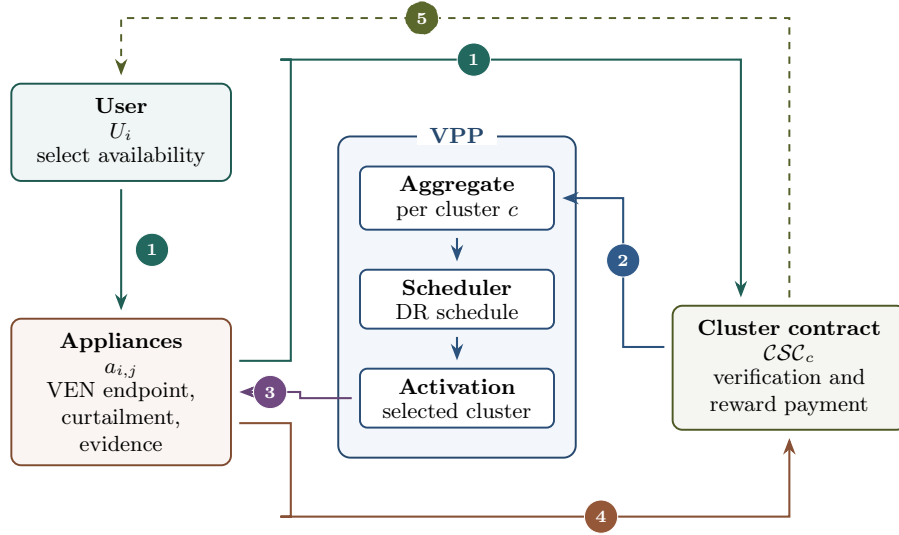


Fig. 1. Architecture-level view of the recurring scheduling phase without a gateway.

- *Step 1 - Availability submission:* Each user U_i specifies the daily curtailment availability of their appliances $a_{i,j}$.

- *Step 2 - Scheduling:* The VPP aggregates the availability reported for each cluster c and computes the corresponding DR schedule.
- *Step 3 - Activation:* When a DR event must be executed, the VPP activates the corresponding cluster. The activation signal is sent to the appliances $a_{i,j}$ that belong to the activated cluster.
- *Step 4 - Evidence generation:* Each scheduled and activated appliance $a_{i,j}$ performs the requested curtailment according to its operational constraints and generates evidence of activation. This evidence is submitted for verification.
- *Step 5 - Verification and payment:* The submitted activation evidence is verified. If the evidence is valid and the claimed response satisfies the activation requirements, the contract authorizes the corresponding remuneration.

2.3 Adversary Model and Requirements

The adversary model and requirements are presented in this section.

Adversary Model The following entities are considered for the adversary model:

- External observers may inspect public blockchain data, IPFS content identifiers, activation records, evidence submissions, and payments. They may also control ordinary user accounts or appliances, but cannot break cryptographic primitives considered secure or extract keys from honest appliance HSMs.
- The VPP is honest-but-curious with respect to privacy: it follows the protocol but may try to infer user behavior from the data it receives.
- The RA is trusted to perform registration and pseudonym rotation correctly and not to disclose the user-pseudonym mapping. The RA and VPP are assumed not to collude.
- Manufacturers and OpenADR certificate authorities are trusted to certify genuine appliances and VEN credentials.

The privacy guarantees of the proposed protocol concern application-layer information. Authenticated and encrypted communication channels can prevent a network adversary from reading or undetectably modifying protected protocol messages, while the freshness and replay checks described in Section 3 prevent reuse of previously valid messages. However, transport-layer protection does not hide metadata such as source IP addresses, communication timing, or traffic volume. A sufficiently powerful network observer could therefore use traffic analysis to weaken the unlinkability guarantees of P2 and P3. In deployments where this threat is relevant, the protocol can be combined with network-layer mechanisms such as VPNs or privacy-preserving relays to reduce source correlation. Resistance to global traffic analysis and denial-of-service attacks based on message dropping or delaying remains outside the scope of this work.

Requirements The proposed system must satisfy a set of requirements. We classify these requirements into three groups: functional requirements (F), privacy requirements (P), and security requirements (S).

- F1) *Registration and clustering* Users must be able to register themselves and enroll controllable appliances in the DR program, and each registered appliance must be assigned to a flexibility cluster according to its technical and operational parameters.
- F2) *Scheduling* The VPP must be able to compute cluster-level availability to perform the scheduling [12].
- P1) *Availability confidentiality*. External observers must not learn the availability schedule of an appliance.
- P2) *User-availability unlinkability*. External observers and the VPP must not be able to link a real user identity to a submitted availability.
- P3) *Appliance activation privacy*. Activated appliances must not be linkable to a specific user by unauthorized parties.
- S1) *Activation authenticity and integrity*. Appliances must accept only authentic and integrity-protected activation events issued by the VPP.
- S2) *Replay protection*. Previous activation events and delivery evidence must not be reusable to trigger unauthorized curtailment or duplicate remuneration.
- S3) *Valid evidence*. A valid delivery evidence must be from an authorized appliance, bound to a scheduled activation event and unforgeable.
- S4) *Payment authorization*. Remuneration must be released only once and only to scheduled appliances that submit valid delivery evidence.

3 Privacy-Preserving Demand Response Scheme

The protocol is divided into a *setup phase*, a *registration phase*, and a *operational phase*.

3.1 Setup Phase

This phase is executed once when the system is set up. It initializes the cryptographic infrastructure, deploys the smart contracts, and defines the components required for the correct execution of the protocol.

VPP Initialization The VPP generates an asymmetric key pair (sk_{VPP}, pk_{VPP}) and obtains a certificate $Cert_{VPP}^{CA}$ issued by a trusted Certificate Authority CA. This certificate is used to authenticate the VPP in the OpenADR-based demand response signaling infrastructure, where the VPP acts as a Virtual Top Node (VTN) [2].

RA Initialization The RA generates an asymmetric key pair (sk_{RA}, pk_{RA}) , which is later used to encrypt pseudonyms that need to change. It also possesses a certificate $Cert_{RA}^{CA}$ issued by a trusted authority.

Manufacturer Initialization When a new appliance $a_{i,j}$ is manufactured, it is equipped with a Hardware Secure Module (HSM). The HSM generates an un-exportable private signing key together with its corresponding public key. Thus, each appliance $a_{i,j}$ is provisioned with an EdDSA signing key pair $(sk_{i,j}^{\text{sig}}, pk_{i,j}^{\text{sig}})$.

The public key $pk_{i,j}^{\text{sig}}$ is certified to authenticate the appliance as a Virtual End Node (VEN) with $\text{Cert}_{\text{VEN},i,j}^{\text{CA}}$ issued by an OpenADR Certificate Authority CA. This certificate is later used by the appliance to authenticate itself as an OpenADR Virtual End Node (VEN) when communicating with the VPP, which acts as the Virtual Top Node (VTN).

3.2 Registration Phase

Each user U_i participating in the demand response program interacts with the system through a mobile application. The mobile application is used to create the user account, configure appliance preferences, and assist in the onboarding of new appliances.

Once the user has been authenticated, the VPP authorizes U_i to register appliances through the Registration Authority (RA) and also sends $\text{Cert}_{\text{VPP}}^{\text{CA}}$. From this point onward, communications between U_i , the RA, and the VPP are performed through authenticated secure channels.

Appliance Registration When a new appliance $a_{i,j}$ is installed, the user executes a secure onboarding procedure through the mobile application.

First, the appliance $a_{i,j}$ enters *pairing mode* after first boot or after a factory reset. Then, U_i , through the mobile application, securely connects with $a_{i,j}$ and performs the following steps:

1. Collects information required to contact the VPP as well as $\text{Cert}_{\text{VPP}}^{\text{CA}}$, and the $\text{Cert}_{\text{RA}}^{\text{CA}}$.
2. Defines the appliance parameter vector $\theta_{i,j} = (\text{max}_{i,j}, \text{rec}_{i,j})$.
3. Sends $\theta_{i,j}, \text{Cert}_{\text{VPP}}^{\text{CA}}, \text{Cert}_{\text{RA}}^{\text{CA}}$ to $a_{i,j}$.

After the local pairing procedure, $a_{i,j}$ performs the following operations to obtain the first pseudonym $r = 0$. This pseudonym is renewed in the Operational Phase:

1. Stores the VPP and RA communication parameters.
2. Generates, inside its HSM, a fresh operational signing key pair $(sk_{i,j,r}^{\text{op}}, pk_{i,j,r}^{\text{op}})$.
3. Generates a blockchain address $\text{Addr}_{i,j,r}$.
4. Constructs the registration message $m_{i,j,r}^{\text{reg}} = [P_{i,j}, \theta_{i,j}, pk_{i,j,r}^{\text{op}}, \text{Addr}_{i,j,r}]$.
5. Signs the registration message: $\sigma_{i,j,r}^{\text{reg}} = \text{Sign}_{sk_{i,j}^{\text{sig}}}(m_{i,j,r}^{\text{reg}})$.
6. Sends $(\text{Cert}_{\text{VEN},i,j}^{\text{CA}}, m_{i,j,r}^{\text{reg}}, \sigma_{i,j,r}^{\text{reg}})$ to the RA.

Upon receiving the appliance registration information, the RA performs the following steps:

1. Verifies the validity of the OpenADR VEN certificate $\text{Cert}_{\text{VEN},i,j}^{\text{CA}}$.
2. Verifies the appliance signature: $\text{Verify}_{pk_{i,j}^{\text{sig}}} (m_{i,j,r}^{\text{reg}}, \sigma_{i,j,r}^{\text{reg}}) = 1$.
3. Checks that $\text{Addr}_{i,j,r}$ and $pk_{i,j,r}^{\text{op}}$ have not been previously enrolled, and marks them as used in its local registry.
4. Assigns $a_{i,j}$ to a cluster according to its technical and operational parameters: $C_c \leftarrow \text{AssignCluster}(P_{i,j}, \theta_{i,j})$.
5. Registers $\text{Addr}_{i,j,r}$ and $pk_{i,j,r}^{\text{op}}$ as an authorized operational pseudonym for cluster c in the corresponding $\mathcal{CSC}_c$.
6. Sends the assigned cluster c , the address of $\mathcal{CSC}_c$, and the necessary OpenADR registration parameters to $a_{i,j}$ and U_i .

At the end of this phase, $a_{i,j}$ is cryptographically bound to U_i , assigned to a cluster c , registered as an OpenADR VEN, and authorized to interact with the corresponding $\mathcal{CSC}_c$ under the pseudonym $(\text{Addr}_{i,j,r}, pk_{i,j,r}^{\text{op}})$.

Credential lifecycle Operational appliance keys are renewed through the pseudonym-rotation procedure described in Section 3.3. Long-term VPP, RA, and appliance certificates follow the corresponding CA renewal and revocation procedures. If an appliance credential is compromised, its operational pseudonym is deauthorized and the appliance must be securely reprovisioned and re-enrolled.

3.3 Operational Phase

This daily procedure is executed to collect appliance availability and support the scheduling optimization performed by the VPP. The protocol is executed independently for each registered appliance $a_{i,j}$ before the scheduling decision for day d is computed. The number of time slots in one day is defined as T .

Step 1 - Availability Submission The user U_i , through the mobile application, specifies the availability of $a_{i,j}$ for day d :

1. Selects the availability schedule of $a_{i,j}$ for day d .
2. Constructs the availability bitmap $b_{i,j,d} \in \{0,1\}^T$, where 1 indicates that $a_{i,j}$ is available for curtailment during time slot t , and 0 indicates that it is not available.
3. Sends $b_{i,j,d}$ to $a_{i,j}$.

When $a_{i,j}$ receives the information, it performs the following steps:

1. Stores $b_{i,j,d}$ locally and encrypts it under the VPP public key pk_{VPP} using a probabilistic public-key encryption scheme $B_{i,j,d} = \text{Enc}_{pk_{\text{VPP}}}(b_{i,j,d})$.
2. Uploads $B_{i,j,d}$ to IPFS and obtains the content identifier $\text{cid}_{i,j,d}$.
3. Using the blockchain address $\text{Addr}_{i,j,r}$ assigned during appliance registration, submits $(d, \text{cid}_{i,j,d})$ to $\mathcal{CSC}_c$.

Step 2 - Scheduling For each cluster c and day d , the VPP computes the aggregate availability and determines the activation intervals. The VPP performs the following operations:

1. Retrieves from the CSC_c all availability submissions registered for day d .
2. Fetches from IPFS all the encrypted payloads referenced by $\text{cid}_{i,j,d}$.
3. Decrypts each encrypted payload using sk_{VPP} and recovers the corresponding set of availability bitmaps $b_{i,j,d}$.
4. Computes the number of available $a_{i,j}$ in each t $N_{c,d}[t] = \sum_{a_{i,j} \in c} b_{i,j,d}[t]$.
5. Computes the aggregate curtailable power available in each time slot $A_{c,d}[t] = N_{c,d}[t] \cdot P_c$.

After computing the aggregate availability, the VPP solves the scheduling optimization problem [12] according to the market request, the available cluster capacity, and the operational constraints of the appliances. The optimization problem is outside the scope of this work.

Let $\mathcal{I}_{c,d}$ denote the set of intervals scheduled for cluster c on day d , which is the output of the scheduling problem. Each scheduled interval is represented as $I = \{t_{\text{ini}}, t_{\text{fin}}\}$. For each scheduled interval $I \in \mathcal{I}_{c,d}$, the VPP performs the following steps:

1. Selects the set of appliance addresses that are available during the whole interval $\alpha_{c,d,I} = \{\text{Addr}_{i,j,r} : b_{i,j,d}[t] = 1, \forall t \in I\}$.
2. Registers $\alpha_{c,d,I}$ in CSC_c : $\text{Schedule}_{c,d,I} = (d, I, \alpha_{c,d,I})$.

Step 3 - Activation When the scheduled flexibility must be activated, the VPP acts as the OpenADR VTN and issues an OpenADR event directly to the registered appliance VENs. For each activated interval I of cluster c , the VPP performs the following steps:

1. Generates a fresh OpenADR event identifier eventID , associated with the activation event.
2. Defines the activation event metadata: $e_{c,d,I} = (c, d, I, \text{eventID})$.
3. Constructs an OpenADR event message as a VTN:

$$\text{OADR}_{c,d,I}^{\text{evt}} = (\text{eventID}, \text{modificationNumber}, \text{marketContext}, \text{eiActivePeriod}(I), \text{eiEventSignal}(s), \text{eiTarget}(c)).$$

4. Sends the OpenADR event directly to the registered appliance VENs using the OpenADR event-distribution service: $\text{oadrDistributeEvent}(\text{OADR}_{c,d,I}^{\text{evt}})$.
5. Computes a hash of the OpenADR event: $h_{c,d,I}^{\text{evt}} = H(\text{OADR}_{c,d,I}^{\text{evt}})$.
6. Records the activation in the CSC_c by storing: $(\text{eventID}, d, I, h_{c,d,I}^{\text{evt}})$.

Upon receiving the OpenADR event, each $a_{i,j}$ acts as a VEN and processes the event according to the OpenADR event-handling procedure. In particular, $a_{i,j}$ performs the following steps:

1. Verifies the authenticity of the VTN through $\text{Cert}_{\text{VPP}}^{\text{CA}}$.
2. Checks that the OpenADR `eventID` has not already been processed.
3. Extracts $\text{eiTarget}(c)$ and verifies that it matches the locally stored cluster assignment.
4. Checks that its $\text{Addr}_{i,j,r}$ belongs to $\alpha_{c,d,I}$ recorded in $\mathcal{CSC}_c$.
5. Verifies that the activation interval I is compatible with its locally stored availability bitmap $b_{i,j,d}$.
6. If all checks succeed, curtails its power consumption for the full duration of interval I .

Step 4 - Evidence After the activation interval I has finished, each appliance $a_{i,j}$ that has successfully curtailed its consumption for the full duration of the interval generates a delivery attestation. This attestation is used to prove that $a_{i,j}$ participated in the activation event and to enable subsequent verification and reward allocation. If an appliance fails to execute the requested curtailment, it cannot produce valid delivery evidence and is therefore not remunerated. Each appliance $a_{i,j}$ performs the following steps:

1. Verifies that it remained curtailed during the activation interval I .
2. Generates an operational key pair $(sk_{i,j}^{\text{op},r+1}, pk_{i,j}^{\text{op},r+1})$ inside its HSM and derives a blockchain address $\text{Addr}_{i,j}^{r+1}$. The appliance then constructs a rotation request encrypted under the RA public encryption key $C_{i,j,d,I}^{\text{rot}} = \text{Enc}_{pk_{\text{RA}}}(\text{Addr}_{i,j}^{r+1}, pk_{i,j}^{\text{op},r+1})$.
3. Constructs the delivery attestation message

$$m_{i,j,d,I}^{\text{del}} = (\text{eventID}, d, I, \text{Addr}_{i,j,r}, C_{i,j,d,I}^{\text{rot}}).$$

4. Computes the corresponding attestation digest and signs it

$$h_{i,j,d,I}^{\text{del}} = H(m_{i,j,d,I}^{\text{del}}), \quad \sigma_{i,j,d,I}^{\text{del}} = \text{Sign}_{sk_{i,j,r}^{\text{op}}}(h_{i,j,d,I}^{\text{del}}).$$

5. Submits the evidence tuple

$$\pi_{i,j,d,I}^{\text{del}} = (m_{i,j,d,I}^{\text{del}}, h_{i,j,d,I}^{\text{del}}, \sigma_{i,j,d,I}^{\text{del}})$$

directly to the corresponding cluster smart contract $\mathcal{CSC}_c$ using the appliance blockchain address $\text{Addr}_{i,j,r}$.

Step 5 - Payment After receiving $\pi_{i,j,d,I}^{\text{del}}$, $\mathcal{CSC}_c$ performs the following steps:

1. Checks that an activation record exists for `eventID`, day d , and interval I .
2. Checks that $\text{Addr}_{i,j,r}$ belongs to the scheduled set $\alpha_{c,d,I}$.
3. Verifies that $m_{i,j,d,I}^{\text{del}}$ refers to the expected activation event, day, interval, and appliance address.
4. Retrieves the operational public key $pk_{i,j,r}^{\text{op}}$ registered for $\text{Addr}_{i,j,r}$.
5. Verifies the appliance signature: $\text{Verify}_{pk_{i,j,r}^{\text{op}}}(h_{i,j,d,I}^{\text{del}}, \sigma_{i,j,d,I}^{\text{del}})$.
6. Checks that $\text{Addr}_{i,j,r}$ has not already been paid for the same activation event.
7. Credits the predefined reward to the appliance address $\text{Addr}_{i,j,r}$.
8. Marks $\text{Addr}_{i,j,r}$ as paid for `eventID`.

Req.	Main mechanism	Section 3
P1	Encryption of availability before public storage	Step 1
P2	Operational pseudonyms	Registration, Step 1
P3	Operational pseudonyms, pseudonym rotation	Steps 3–5, rotation
S1	VPP authentication, event hash	Step 3
S2	Fresh event identifiers and one-payment state	Steps 3 and 5
S3	Registered operational keys and signed delivery evidence	Registration, Steps 4–5
S4	Schedule membership, evidence verification	Step 5

Table 1. Traceability between requirements and protocol mechanisms.

Pseudonym rotation The RA periodically retrieves pending rotation ciphertexts and performs the following steps:

1. Decrypts them using sk_{RA} and obtains $(Addr_{i,j}^{r+1}, pk_{i,j}^{op,r+1})$.
2. Verifies that the old address $Addr_{i,j}^r$ is currently authorized in CSC_c , that the new address $Addr_{i,j}^{r+1}$ has not been previously enrolled, and that the request refers to a valid accepted delivery event.
3. If all checks succeed, the RA authorizes $Addr_{i,j}^{r+1}$ and $pk_{i,j}^{op,r+1}$ in CSC_c for the same cluster c .
4. The old address $Addr_{i,j}^r$ is then deactivated for future scheduling epochs.

The RA performs authorizations and deactivations in batches and does not publish an explicit old and new mapping on chain.

4 Discussion

This section evaluates the proposed architecture with respect to the functional, privacy, and security requirements defined in Section 2.3.

Table 1 summarizes where the main mechanisms supporting each privacy and security requirement are introduced in the protocol.

4.1 Functional Feasibility

The proposed architecture provides the functional capabilities required for appliance DR by relying on the protocol phases defined in Section 3. Requirement F1 is addressed by the setup and registration procedures described in Sections 3.1 and 3.2, where the necessary cryptographic material, appliance credentials, operational pseudonyms, and cluster memberships are established before any DR operation is executed. These procedures ensure that only valid appliances are enrolled and that each appliance is associated with the appropriate cluster smart contract.

Requirement F2 is addressed by the scheduling phase described in Section 3.3. In particular, the availability-submission and scheduling steps enable the VPP to derive cluster-level flexibility from appliance availability reports, while the activation step allows the VPP to issue DR events to the selected appliances. The subsequent evidence and payment step provides the mechanism through

which completed activations are verified and remunerated. Therefore, the protocol phases jointly provide the registration, clustering, scheduling, activation, verification, and payment functionality required to support the privacy and security properties analyzed below.

4.2 Security and Privacy Discussion

P1 - Availability confidentiality. P1 requires that external observers do not learn the availability submitted by an appliance. This is achieved by separating the public availability record from its content. The blockchain stores only the submission day, the operational pseudonym, and the IPFS content identifier, whereas the availability bitmap is encrypted under pk_{VPP} using probabilistic public-key encryption before being uploaded to IPFS. Therefore, public blockchain observers, other users, other appliances, and unauthorized third parties may observe that a pseudonymous appliance submitted availability, but cannot recover the submitted time slots. Thus, P1 is satisfied under the confidentiality of the encryption scheme.

P2 - User-availability unlinkability. P2 requires that external observers and the VPP cannot link a real user identity to a submitted availability report. For external observers, this follows from the use of operational pseudonyms: availability submissions are recorded in CSC_c under appliance addresses rather than real identities, and the user-pseudonym mapping is not published on chain. The VPP has stronger access because it decrypts availability bitmaps to compute the schedule. Hence, the protocol does not hide appliance-level availability from the VPP; instead, it limits the VPP to pseudonym-level availability knowledge. Since the user-pseudonym mapping is held by the RA and the VPP and RA are assumed not to collude, the VPP cannot directly associate a submitted availability bitmap with a real user. Therefore, P2 is satisfied within the stated trust model.

P3 - Appliance activation privacy. P3 requires that activated appliances are not linkable to specific users by unauthorized parties. Activation records, scheduled sets, evidence submissions, and payments are expressed through operational pseudonyms rather than real user identities. Consequently, external observers may learn that a given pseudonym participated in an activation, but cannot directly infer the user behind it. The VPP necessarily knows the activated cluster and the scheduled pseudonyms, but does not possess the RA-held user-pseudonym mapping. Pseudonym rotation further reduces long-term linkability: the new address and operational public key are encrypted under pk_{RA} and processed by the RA without publishing an explicit old-new mapping. Rotations are processed in batches to reduce timing and cardinality leakage. Thus, P3 is satisfied against external observers and an honest-but-curious VPP under the non-collusion assumption.

S1 - Activation authenticity and integrity. S1 requires appliances to accept only authentic and integrity-protected activation events issued by the VPP. The VPP acts as the OpenADR VTN and authenticates with $Cert_{VPP}^{CA}$. Upon receiving an event, the appliance verifies the VTN certificate, checks that the *eventID* has not already been processed, verifies that the target cluster matches its local cluster assignment, and checks that its operational address belongs to the scheduled set $\alpha_{c,d,I}$ stored in CSC_c . Integrity is additionally supported by the on-chain activation hash: the appliance verifies that $H(OADR_{c,d,I}^{evt}) = h_{c,d,I}^{evt}$. These checks bind the activation to the certified VPP, the scheduled interval, the target cluster, and the on-chain schedule, preventing forged or modified activation events unless VPP credentials or cryptographic primitives are compromised.

S2 - Replay protection. S2 requires that previous activation events and delivery evidence cannot be reused to trigger unauthorized curtailment or duplicate remuneration. Activation replay is mitigated because each OpenADR event contains an *eventID*, *d* and *I*, and appliances reject identifiers that have already been processed. Evidence replay is prevented by CSC_c , since each delivery attestation is bound to a specific *eventID*, day *d*, interval *I*, and operational address $Addr_{i,j,r}$. The contract also checks that the submitting address has not already been paid for the same event and marks it as paid after successful verification. Therefore, replayed activation messages and reused delivery attestations are rejected.

S3 - Valid evidence. S3 requires delivery evidence to be signed with an authorized appliance key and bound to a scheduled activation event. During registration, the RA registers $pk_{i,j,r}^{op}$ and $Addr_{i,j,r}$ in the corresponding CSC_c . After an activation interval, the appliance signs a delivery attestation containing the *eventID*, day, interval, operational address, and optional encrypted rotation request. The contract accepts the evidence only if the claimed activation exists, the submitting address belongs to $\alpha_{c,d,I}$, the attestation refers to the expected event, $h_{i,j,d,I}^{del} = H(m_{i,j,d,I}^{del})$, and the signature verifies under the registered operational public key. These checks reject evidence from unauthorized keys, unscheduled pseudonyms, or different activation events. The link between signed evidence and physical curtailment relies on the trusted appliance control and measurement component connected to the HSM.

S4 - Payment authorization. S4 requires remuneration to be released only once and only to scheduled appliances that submit valid delivery evidence. This is enforced by CSC_c : before authorizing payment, the contract verifies that the activation record exists, the submitting address belongs to the scheduled set, the evidence refers to the expected event, the hash and signature are valid, and the address has not already been paid for the same activation. If any check fails, the evidence is rejected and no remuneration is released. Otherwise, the reward is credited to the operational address and that address is marked as paid for the event.

5 Conclusions

This paper presented a privacy-preserving and secure architecture for appliance-level demand response in Virtual Power Plants. The proposed scheme supports appliance registration, clustering, availability reporting, activation, evidence verification, and payment through the combined use of OpenADR, operational pseudonyms, cryptographic signatures, encrypted availability reports, and cluster smart contracts.

The architecture enables the VPP to compute aggregate cluster-level flexibility and activate scheduled appliances, while limiting the disclosure of user-appliance relationships to unauthorized actors. Moreover, signed activation evidence allows the smart contract to verify participation before remuneration, reducing the risk of forged compensation claims. The discussion shows that the proposed protocol satisfies the functional, privacy, and security requirements defined for the system.

Future work will focus on implementing the architecture, evaluating its computational and communication costs, and integrating the protocol with concrete VPP scheduling algorithms.

Acknowledgments. This research is supported by Project PID2024-156914OB-C42 “SAFE/DATA”, funded by MICIU/AEI/ 10.13039/501100011033 and FEDER, UE. Joan Ferré-Queralt is supported by the Spanish Government under an FPU grant (ref. FPU23/01275). Alexandre Viejo is a Serra-Hünter Fellow.

References

1. Adhikari, R., Pipattanasomporn, M., Rahman, S.: An algorithm for optimal management of aggregated hvac power demand using smart thermostats. *Applied Energy* **217**, 166–177 (2018)
2. Alliance, O.: Openadr 2.0 b profile specification. Raportti. OpenADR Alliance (2013)
3. Di Silvestre, M.L., Gallo, P., Sanseverino, E.R., Sciume, G., Zizzo, G.: Aggregation and remuneration in demand response with a blockchain-based framework. *IEEE Transactions on Industry Applications* **56**(4), 4248–4257 (2020)
4. Erdinc, O., Taşçıkaraoğlu, A., Paterakis, N.G., Eren, Y., Catalão, J.P.: End-user comfort oriented day-ahead planning for responsive residential hvac demand aggregation considering weather forecasts. *IEEE Transactions on Smart Grid* **8**(1), 362–372 (2016)
5. Gehbauer, C., Lee, E.S., Wang, T.: An evaluation of the demand response potential of integrated dynamic window and hvac systems. *Energy and Buildings* **298**, 113481 (2023)
6. Kirby, B.J.: Load response fundamentally matches power system reliability requirements. In: 2007 IEEE power engineering society general meeting. pp. 1–6. IEEE (2007)
7. Kundur, P., et al.: Power system stability. *Power system stability and control* **10**(1), 7–1 (2007)

8. Li, H., Lin, X., Yang, H., Liang, X., Lu, R., Shen, X.: Eppdr: An efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid. *IEEE Transactions on Parallel and Distributed Systems* **25**(8), 2053–2064 (2013)
9. Ma, R., Yi, Z., Xiang, Y., Shi, D., Xu, C., Wu, H.: A blockchain-enabled demand management and control framework driven by deep reinforcement learning. *IEEE Transactions on Industrial Electronics* **70**(1), 430–440 (2022)
10. Ming, H., Xie, L., Campi, M.C., Garatti, S., Kumar, P.: Scenario-based economic dispatch with uncertain demand response. *IEEE Transactions on Smart Grid* **10**(2), 1858–1868 (2017)
11. Njoka, G.M., Mogaka, L., Wangai, A.: Impact of variable renewable energy sources on the power system frequency stability and system inertia. *Energy Reports* **12**, 4983–4997 (2024)
12. Orduz, N., Tsuji, T., Ferré-Queralt, J., Castellà-Roca, J., Viejo, A.: A technical score and cryptography enabled framework for multi market virtual power plant demand response. In: 32nd International Council on Electrical Engineering Conference (ICEE 2026). Seoul, Korea (2026)
13. Parvania, M., Fotuhi-Firuzabad, M., Shahidehpour, M.: Optimal demand response aggregation in wholesale electricity markets. *IEEE transactions on smart grid* **4**(4), 1957–1965 (2013)
14. Parvania, M., Fotuhi-Firuzabad, M., Shahidehpour, M.: Iso’s optimal strategies for scheduling the hourly demand response in day-ahead markets. *IEEE Transactions on Power Systems* **29**(6), 2636–2645 (2014)
15. Rottondi, C., Verticale, G.: Privacy-friendly load scheduling of deferrable and interruptible domestic appliances in smart grids. *Computer Communications* **58**, 29–39 (2015)
16. Singh, S., Kumar, A.: Demand response program solution to manage congestion in transmission network considering uncertainty of load. In: 2017 8th international conference on computing, communication and networking technologies (ICCCNT). pp. 1–6. IEEE (2017)
17. Venizelou, V., Tsolakis, A.C., Evagorou, D., Patsonakis, C., Koskinas, I., Therapontos, P., Zyglakis, L., Ioannidis, D., Makrides, G., Tzovaras, D., et al.: Dso-aggregator demand response cooperation framework towards reliable, fair and secure flexibility dispatch. *Energies* **16**(6), 2815 (2023)
18. Wei, M., Zhong, J.: Optimal bidding strategy for demand response aggregator in day-ahead markets via stochastic programming and robust optimization. In: 2015 12th International Conference on the European Energy Market (EEM). pp. 1–5. IEEE (2015)
19. Yang, Q., Wang, H., Wang, T., Zhang, S., Wu, X., Wang, H.: Blockchain-based decentralized energy management platform for residential distributed energy resources in a virtual power plant. *Applied Energy* **294**, 117026 (2021)
20. Yi, Z., Xu, Y., Gu, W., Wu, W.: A multi-time-scale economic scheduling strategy for virtual power plant based on deferrable loads aggregation and disaggregation. *IEEE Transactions on Sustainable Energy* **11**(3), 1332–1346 (2019)
21. Yolcan, O.O.: World energy outlook and state of renewable energy: 10-year evaluation. *Innovation and Green Development* **2**(4), 100070 (2023)
22. Yu, H., Zhang, J., Ma, J., Chen, C., Geng, G., Jiang, Q.: Privacy-preserving demand response of aggregated residential load. *Applied Energy* **339**, 121018 (2023)
23. Zhang, X., Hug, G.: Bidding strategy in energy and spinning reserve markets for aluminum smelters’ demand response. In: 2015 IEEE power & energy society innovative smart grid technologies conference (ISGT). pp. 1–5. IEEE (2015)

Hidden Figures: Fingerprinting Mobile Phones through Passive Background App Traffic

Shiva Azizzadeh¹[0000–0003–1330–1921] and Gunes Acar¹[0000–0002–1621–8333]

Radboud University, Nijmegen, The Netherlands
shiva.azizzadeh@ru.nl
gacar@cs.ru.nl

Abstract. Traffic analysis has been used for over two decades to extract sensitive information from network communications. Most existing approaches rely on user-generated activity, such as web browsing, which introduces constraints on the attack scenario and requires assumptions about user behavior. These limitations raise the question of whether similar attacks remain feasible when no user interaction is present. In this work, we propose a passive fingerprinting attack that solely relies on background smartphone network traffic, generated without user involvement. More precisely, we show that communication endpoints extracted from smartphone background traffic contains sufficient information to distinguish user profiles and link device activity across locations. To this end, we collect 400 hours of Wi-Fi traffic representing various user profiles. We identify factors influencing the success of the attack and optimize the attack accordingly. Our results show that an adversary can successfully distinguish individual users with a best-case $F1$ score of 0.89 and accuracy of 0.99 (mean $F1=0.72$, mean accuracy=0.97). We achieve this performance without complex machine learning models or extensive training phases. Instead, we use lightweight distance-based comparisons, which makes the attack practical and efficient. This leads to a straightforward attack setup that enables an adversary to perform user tracking, even in the absence of interaction and despite common privacy protections.

Keywords: Fingerprinting · Traffic analysis · Mobile phones security.

1 Introduction

Mobile phone users expose valuable personal and behavioral data to network eavesdroppers when using online services [2]. Such data can be leveraged for various purposes such as identifying communication patterns in network traffic [20, 60] or detecting malware through application behavior [28]. Notably, even encrypted traffic leaks fine grained information in realistic settings [51, 58]; it can expose user actions and identify applications [61, 30], reveal users’ daily activity patterns and service requirements [32], and localize user movement [50]. Devices also remain distinguishable through transmission characteristics [26] and

communication derived “fingerprints” [62, 2] e.g., web sessions, sets of installed applications, or application behavior [9] even in the presence of encryption [56].

These attacks all rely on network traffic triggered by user interaction, and often assume active users or controlled environments, limiting their applicability in passive or real-world settings [31]. However, mobile devices generate traffic even when users are inactive, due to background activities such as updates, synchronization and telemetry. This raises the question of whether passive background traffic alone can reveal sufficient information for tracking and fingerprinting, as it is lower in volume and variability and therefore harder to exploit. Moreover, while modern MAC address randomization assigns different identifiers per network and defeats traditional identifier-based cross-location tracking, it does not fully prevent device identification, as devices can still be fingerprinted based on wireless metadata and protocol characteristics [36, 57].

In this paper, we present a traffic fingerprinting attack that requires no user interaction. Using multiple user profiles with diverse application sets and different devices, we collect traffic in a realistic Wi-Fi setting, and using only the extracted traffic metadata, we show that fingerprinting remains feasible without any user activity, and this enables user tracking across locations and over time, even if users randomize their MAC addresses [11, 23, 15].

The attack is particularly concerning in shared public networks such as malls or university campuses, where a passive adversary can link observations collected at different locations and times. The potential victims are ordinary users who might move between independent networks, such as from train station to a university. The attack exposes users’ movement patterns, repeated visits, and behavioral routines without user interaction or awareness. The adversary can be a passive network eavesdropper that aims to track employee movements across the corporate network or organizations that track people participating in protests.

Existing passive fingerprinting approaches are either based on timing in which the device is identified based on inter-arrival times of frames or clock skew [40, 55], or classifiers of traffic to recognize the flow produced by applications [52]. However, both groups need to have the target enrolled beforehand to be able to decide in a close-world classification and the question we aim to answer has remained open whether two captures recorded at different times and places can be linked to the same person without any prior signature of that person. Moreover, in contrast to existing traffic analysis approaches that rely on machine learning (ML) and deep learning techniques [42, 46, 45, 38], our attack relies on a simple distance metric and communication endpoints, and it does not require computing resources and costly retraining due to concept drift (shortcomings of many ML-based attacks [27]). Specifically, we make the following contributions:

- We introduce passive background traffic fingerprinting, a traffic analysis attack that does not require user interaction and relies only on smartphone background traffic.
- We conduct a comprehensive experimental evaluation using more than 400 hours of traffic and identify factors influencing the attack. Our attack achieves $F1$ score of 0.89 and an accuracy of 0.99 in a closed-world setting.

- Unlike similar attacks that rely on complex machine-learning models, our attack uses simple distance-based metrics, is extremely lightweight, and does not require extensive traffic data.

2 Preliminaries

We study the feasibility of a traffic analysis attack that identifies similarities between user devices across observations, and uses passive background traffic recorded at multiple locations and times. We first state the problem and then provide the necessary technical background underlying the attack.

2.1 Problem Statement

We consider a set of WiFi networks at different physical locations, each serving a dynamic set of mobile users who connect for limited durations. Over time, a user might connect to multiple networks, forming a history of connections. Due to MAC address randomization, persistent link-layer identifiers might be unavailable, so the adversary cannot associate sessions or distinguish devices by their MAC address. Although a randomized MAC address may stay stable within a single network depending on the randomization mode, this stability is not guaranteed, as platforms such as iOS periodically rotate it [7], and a device uses a different (randomized) MAC address on each network, preventing cross-location tracking at the link layer. This limitation motivates the need for alternative techniques based on traffic analysis, which we present in this paper.

For a given network and observation interval, the *user set* is the set of all devices communicating during that observation period. We model each user device by a profile $u \in \mathcal{U}$, characterized by three significant attributes: device type, operating system, and installed applications. These attributes influence the background communication behavior of the device. Our attack focuses on *passive background* traffic, which is packets a device sends and receives autonomously without direct user interaction. We consider the traffic to be encrypted, so the adversary can only access metadata such as IP addresses of endpoints (§ 2.2).

2.2 Traffic Analysis Background

Traffic analysis attacks infer sensitive information from the observable characteristics of network transmissions, rather than their content [42, 41, 38]. We focus on passive fingerprinting [46, 42], where the adversary monitors traffic without interfering. Even under encryption, exploitable metadata such as packet headers and derived features remains available for traffic analysis, which the adversary assigns to individual profiles to eventually distinguish users. We describe the most relevant metadata features below. We note that availability of these features may differ on anonymous communication networks such as Tor [19].

IP Addresses. Source and destination IPs indicate connection endpoints. Factors such as load balancing, distributed infrastructures, or address changes over time may cause different IP addresses for the same hosts.

Autonomous Systems. Every endpoint belongs to an autonomous system (AS) governing inter-provider routing. An AS spans many IP addresses, enabling the adversary to map dynamic IP addresses to a relatively stable AS number.

Ports. Ports act as gateways that direct data correctly between devices. Client-side ports are ephemeral, but the server-side (well-known) ports indicate the service type (e.g., 443 for TLS) and are useful to the adversary (§5.3).

Additional Information. Packet sizes, timings, directions, and unencrypted headers have been used in prior (often ML-based) attacks [13, 24, 44, 12, 38].

2.3 Android Specifications

We focus on Android devices, whose operating system and vendors’ implementations influence the traffic, both with and without user interaction. For example, Android’s battery management system can limit background network access and periodically terminate background applications through a Least Recently Used (LRU) mechanism [5]. In a low power state like Doze mode, background execution is further restricted and applications cannot initiate arbitrary activities, and this is also affected by their startup state [3]. However, the system may wake processes and grant network access during periodic maintenance windows [4].

3 Attack

The adversary aims to link network traces of the same user device across different physical locations using passive background traffic generated without user interaction such as periodic synchronization. This restriction is intentional as it limits the identification signal to the ones contributed by the activities of the installed applications and filters out user behavior. The adversary collects traces at multiple locations and times, each capturing a device’s network activity within a time window. In an offline phase, the adversary compares traces from different locations by applying a distance metric to pairs of trace representations and comparing the result against a threshold; a distance below the threshold indicates that traces originated from the same device. Matching related traces lets the adversary track devices across locations and infer behavioral patterns over time. We next describe the threat model and the attack phases.

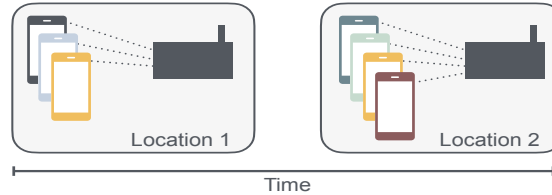


Fig. 1. The adversary records traffic traces in multiple locations (networks) and tries to link users’ traces across locations.

3.1 Threat Model

Assumptions and Adversary Capabilities. The adversary passively monitors the wireless traffic of mobile devices in public environments such as cafés or other open access WiFi deployments, either by sniffing open, unencrypted networks or by deploying a malicious access point. Traffic collection occurs at multiple locations and times. The adversary captures network-layer metadata such as endpoint IP addresses, which is not protected by link-layer encryption (e.g., WPA2), as we assume an open network. We note that *not* all such metadata is persistent or identifying. MAC addresses, for instance, may be randomized or periodically rotated and thus cannot serve as persistent identifiers.

3.2 Attack Workflow

The attack has two main phases: data collection and trace linking.

Collection Phase. The adversary monitors a WiFi access point at two or more locations, passively recording all traffic from connected devices during a limited period (Figure 1). Each observed device is distinguished by a transient identifier derived from its MAC address during a single session. If MAC address rotation (due to randomization) happens mid-session, the adversary may attempt to relink the new MAC to the same device by checking the continuity in the 802.11 sequence numbers. When randomization is enabled, a different MAC address is used for each session and location (§2.1), so identification based on them across sessions and locations is not possible. We focus on a two location scenario for the clarity of notation: $\mathcal{T}^{(1)} = \{T_1^{(1)}, T_2^{(1)}, \dots, T_{n_1}^{(1)}\}$ and $\mathcal{T}^{(2)} = \{T_1^{(2)}, T_2^{(2)}, \dots, T_{n_2}^{(2)}\}$, where $\mathcal{T}^{(1)}$ and $\mathcal{T}^{(2)}$ denote the traces collected at the first and second locations, respectively. The attack compares each pair of traces independently, so additional locations still reduce to additional pairwise comparisons and do not change the procedure.

Trace linking. The adversary analyzes the collected traffic to link traces of the same device. In preprocessing, captured traffic is transformed into a feature-based metadata representation (§5.1), where each trace is a set of timestamped metadata features from observed flows. For two traces T_i and T_j , the adversary computes a similarity score $\text{sim}(T_i, T_j)$ using a predefined distance over the features (§5.2), and a decision function determines whether they originate from the same device (§5.4). Applied to all pairs, this procedure results in matched pairs that link device sessions across locations and reveal user mobility patterns.

4 Experimental Setup

We build an experimental setup to implement and evaluate the attack, through producing the dataset of traffic traces for different user profiles.

4.1 Setup

The experimental setup consists of a mobile phone and a WiFi access point (Raspberry Pi 3 with `wpa_supplicant` [34]) that captures traffic using `tcpdump` [22]. The first round of experiments (January 2023) used a POCO F2 Pro (Android 10); the second (October 2024) used a Samsung Galaxy A14 5G (Android 14) to test the attack feasibility on a different vendor with a newer Android version.

User Profiles. To represent different users, we create each profile as a distinct set of installed applications, drawn from top 50 apps across 31 Google Play Store categories (sampled December 2022 and September 2023). Each profile’s size is randomly chosen from [50, 100] (mean 80, std 20), and an app may appear in multiple profiles. Each category receives a popularity score from public statistics ([6]) (e.g., download counts, ratings), normalized to a probability distribution. Since the selected categories’ scores sum to less than 1, a profile’s number of apps falls below its target; we fill the remainder with the "communication" category, as mobile phone users often have a wide range of them installed.

Sampling Procedure. A profile is generated by (i) determining its size randomly from [50, 100]; (ii) selecting a weighted random subset of categories by popularity score; (iii) distributing the size across these categories (at least one app per category); (iv) picking random apps from each category according to numbers determined in step (iii); and finally (v) filling the remaining slots with random "communication" apps.

We generate 40 profiles from 473 applications, averaging 75 apps per profile (a few selected apps could not be installed, lowering the average number of apps). Profiles share between 4 and 33 applications, with an average overlap of 16.

App Activation and Traffic Recording. We factory reset the phone, disable as many pre-installed applications as possible, and install all apps needed across profiles. Applications are initially disabled and are activated per profile using `adb`. To capture a profile’s background traffic, we launch its apps once to trigger initialization, ensuring each app completes its cold start (§2.3) so it can later sleep and resume while idle. We then close the apps, turn off the screen, and leave the device connected and idle while recording traffic for one hour. Afterward, the apps are disabled and the process repeats for the next profile.

4.2 Recording Repetitions.

To increase dataset diversity, we record multiple repetitions per profile, randomizing the app activation order each time to mimic realistic behavior of users (different order of using applications). We keep the default power management settings, as most users leave them unchanged. With one hour recording per profile (using `tcpdump`), 40 profiles, and 10 repetitions each, this yields 400 hours of traffic that serve as input for the attack and evaluation.

5 Attack Procedure

Below, we describe the four main steps of the attack: metadata extraction, endpoint processing, uniqueness analysis, and trace comparison.

5.1 Metadata Extraction

The recorded traffic metadata is the attack’s main input. The adversary processes raw traces and extracts the header field information. Although features such as timing and protocol type are available, our attack focuses on connection endpoints, since they can reflect a user’s installed apps and might be user-specific.

5.2 Endpoint Sets and Filtering

For each endpoint IP address in captured data, we resolve its autonomous system (AS) via Maxmind GeoIP database¹, representing each endpoint by its IP address and AS name. Then we remove endpoints with very low traffic volume, which tend to be outliers, using the standard Tukey formula for threshold $t_v = Q_1 - 1.5 * IQR$ [54].

5.3 Endpoint Uniqueness

An endpoint’s distinguishing power depends on how rare it is; an endpoint appearing in all profiles has no distinguishing power. We assign each AS a commonness value, where a lower value indicates higher uniqueness and more significance for the decision. The commonness of an AS is the average of three characteristics: (i) the number of distinct profiles communicating with the AS; (ii) the destination ports these profiles use; and (iii) the AS’s packet size diversity.

When fewer profiles communicate with an AS, the AS can be considered more unique. The fewer ports being used to communicate with an AS shows that the behavior of that AS is more specific. Finally, lower packet-size variability indicates more consistent, distinctive behavior. To measure packet-size diversity of an AS, we categorize its packet sizes into ten bins from 0.1 (least diverse) to 1.0 (most diverse), compute this per AS–profile pair, and average over profiles. Since equal averages can hide different variability, we take a weighted average over the range between the AS’s lowest and highest bins. We prefer this to metrics such as standard deviation because, for behavioral consistency, the extreme bins matter more than their distribution. For example, “Fastly Inc.” is used by 37 profiles, on average 37 for the same purpose (same ports), with average packet-size diversity 7.4, giving a commonness of 27.1; “Opera Software Americas LLC” on the other hand scores 1, 1, and 0.07 respectively, which gives it a commonness value of 0.6.

Uniqueness Decision. The commonness value decides whether an endpoint’s IP or AS information is used in the attack step. To this end, we define a

¹ <https://www.maxmind.com/en/geoip-databases>

commonness threshold $t_c = Q_3 + 1.5 * IQR$ that detects outliers (similar to §5.2) and defines in which cases the AS cannot contribute to a clearer decision. The information of a sufficiently unique AS is kept; otherwise we extend it with port information into AS-port strings (excluding the ubiquitous HTTPS port 443). For example, “Amazon.com, Inc” is used by 37 profiles, all over HTTPS and port 5222, but only one profile over port 123; so the string “Amazon.com, Inc.123” (and other informative ports of the AS) is used instead of the AS name.

5.4 Trace Comparison

After preprocessing (§5.1 to §5.3), each trace is represented by its communication endpoints, and each endpoint can be an IP address, an AS, or the combined AS-port (§5.3). Comparing traces provides information on pairwise similarity that lets the adversary infer which traces are related and track a user across locations over time. This requires two steps: (i) distance calculation, where we compute the Jaccard distance over the presence or absence of endpoints; and (ii) similarity check, where a distance threshold determines whether two traces are related.

Each trace consists of categorical endpoint identifiers that are drawn from a large pool (universe), so the comparison is between two sparse binary vectors, and shared absence here carries no information. In other words, two profiles not contacting a specific AS have nothing meaningful in common. This filters out the metrics that count absences as agreement and are affected by the size of the pool rather than profile’s behavior (such as Hamming or Euclidean). Jaccard distance here can measure what fraction of endpoints contacted in either trace were contacted by both. Over all candidate pairs, a distance lower than the threshold shows the traces are deemed to originate from the same user.

6 Evaluation

In this section, we analyze the performance of the attack using random repetitions and different attack parameters such as similarity threshold.

6.1 Evaluation Procedure

The evaluation procedure consists of two steps which are explained as follows.

Preparation. We obtain the raw dataset through traffic measurements and extend endpoints with AS and port information, producing an extended dataset of labeled (profile, repetition) traces of endpoint information (IP address, AS, AS-port). Mirroring a real attack, this dataset contains both the training information an adversary gathers beforehand, and the traces of unidentified victims.

Random Repetitions. For each repetition: (i) we split the whole 400 traces into training (80%) and testing (20%) sets, and we remove test profiles from the training set to avoid bias. This separation ensures that no trace from the same profile occurs both in train and test sets, which would give an unrealistic advantage to the attacker. (ii) We then choose a random starting point in time for

each trace, and extract a fixed-length time window with a different starting point from all traces. This is done to allow an observation window shorter than one hour, and to use different data for each repetition. (iii) Next, we compute commonness values and find the optimal decision threshold on the training subset. (iv) Finally, we perform pairwise trace comparison on the test subset. It should be noted that training and testing sets are not used in the machine-learning sense, and in our method, no machine learning model is fitted.

6.2 Evaluation Metrics

We classify the decisions as in Table 1 and compute the accuracy $((TP + TN)/(TP + TN + FP + FN))$ for overall correctness, but the dataset is highly imbalanced since only a small fraction of trace pairs are related. Table 2 shows that on average only 7% of trace pairs match, while the remaining 93% are unrelated. Hence, we also use precision, recall, and F_1 score $(2 \cdot (precision \cdot recall)/(precision + recall))$ to evaluate performance.

Table 1. Decision Classes.

Result	Label	Guess	Outcome
$dist > t$	Match	Other	False Negative
$dist > t$	Other	Other	True Negative
$dist < t$	Match	Match	True Positive
$dist < t$	Other	Match	False Positive

Table 2. Average Ground Truth Matches.

	Mean	Std	Min	Max
Related Pairs	141	56	53	273
Unrelated Pairs	2360	360	1662	3027
Total	2506	377	1805	3148

6.3 Factors Influencing the Attack Success

We study how several factors affect the accuracy of the attack, then derive the final attack setup used in the main evaluation.

Recording Duration. The recording duration determines how much information is available. We test 500–3540 seconds with random offsets; longer recordings can expose more endpoints and improve performance (§6.4).

Decision Threshold. We use a threshold to judge whether two traces originate from one user; a misconfigured value inflates false positives or negatives. We divide the range of computed distances into 10 equal intervals, giving 10 candidate thresholds in $[0, 1]$ (the Jaccard range), and select the one maximizing F_1 against the ground truth.

Endpoint Representations. We implement the attack with three endpoint sets: ASes only, IP addresses only, and the combination of ASes and AS-port strings (where an AS is non-unique).

Applications and Devices. To show that the results are not tied to specific profiles or hardware, we repeat the attack on a second phone (Samsung Galaxy A14 5G, Android 14) with new profiles from an updated app list (§4.1).

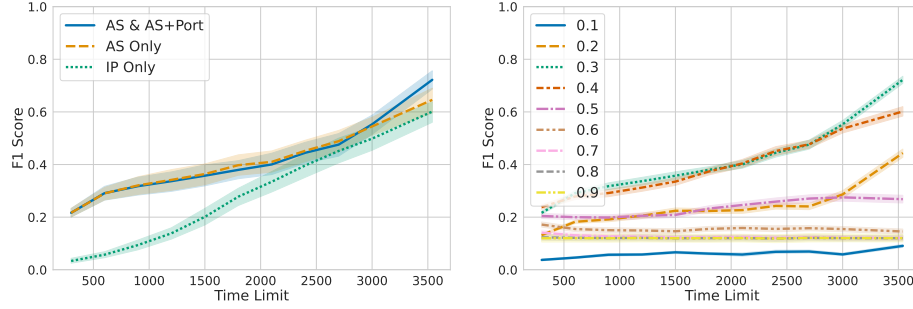


Fig. 2. Left: The plot shows attack performance (F_1 score) in different observation time limits (seconds) and allows us to compare the quality of different types of endpoints. The combination of AS and AS-port strings leads to the best performing attack setting. Right: The plot compares the average F_1 score in different observation time limits (seconds) considering different decision thresholds. It indicates that a threshold of 0.3 leads to the best attack performance in all tested time limits. Other thresholds are either too restrictive or relaxed, causing false positive or false negative decisions.

Table 3. Evaluation of the attack using 30 repetitions.

	5 Minutes				30 Minutes				60 Minutes			
	Mean	Std	Min	Max	Mean	Std	Min	Max	Mean	Std	Min	Max
Accuracy	0.93	0.01	0.90	0.96	0.93	0.01	0.91	0.95	0.97	0.01	0.94	0.99
Precision	0.30	0.10	0.14	0.46	0.38	0.15	0.16	0.68	0.85	0.09	0.67	0.98
Recall	0.18	0.05	0.08	0.30	0.35	0.10	0.14	0.57	0.63	0.14	0.30	0.89
F_1 score	0.21	0.05	0.14	0.34	0.35	0.10	0.17	0.54	0.72	0.09	0.40	0.89

6.4 Experiments

In the first phase, we vary one factor per experiment while fixing the rest, run 30 random repetitions each, and retain the best performing setting. Our main observations regarding the influencing factors are as follows:

Recording time is a tradeoff between attack overhead and success, so rather than an optimal value we report the attack success over a range. Figure 2 (right) shows that across time limits, 0.3 is the optimal decision threshold as it gives the best F_1 score, balancing false positives and negatives.

Table 3 summarizes performance over 30 simulation rounds on the POCO phone for different time limits at the optimal threshold of 0.3. The results show that although the constantly high accuracy (0.93-0.97) reflects data imbalance rather than detection ability, the attack success still improves sharply with the increase of observation time, with the average F_1 score of 0.21 for 5-minute increases to 0.72 for one-hour traces, with the possibility of reaching to 0.89. Figure 2 shows that using IP addresses results in the lowest F_1 score, whereas AS information or AS-port strings achieve higher success.

Finally, Figure 3 (left) summarizes performance for each device alone and for the combination of two phones (cross-device matching); the attack performs

well in all setups. The relative values in the right plot are the normalized average of how many distinct ASes, IP addresses, and packets exists in different observation time limits for each phone over 30 repetitions of the attack. The two plots illustrate that the attack against the second phone (Samsung) with more up-to-date Android version (14 vs 10) achieves higher success due to the larger amount of network traffic sent to/from the phone, providing more signals to the adversary even without user interaction. The same reasoning applies to the higher success rate of the attack in the observations with longer durations, as it provides the attacker with more identifying signals.

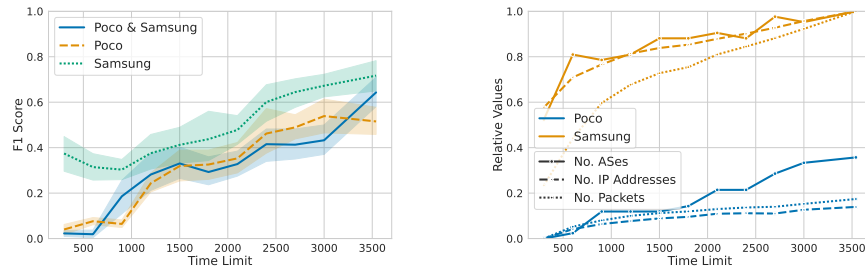


Fig. 3. Left: The comparison of attack performance ($F1$ score) for two devices and their combination within different observation time limits (seconds) shows that the attack feasibility does not rely on the device. Right: Relative values on the y-axis represent the normalized counts of distinct ASes, IP addresses, and packets observed across different time limits (seconds), scaled to the maximum observed value for each metric. Larger volume of endpoints and packets available to the attacker may explain the higher success rate for the Samsung phone (Android 14), and for longer observations.

7 Discussion

Below we discuss the limitations of our work and possible countermeasures.

7.1 Limitations

Our real-world assumptions and chosen setup introduce the following limitations.

Link-Layer Encryption. As the attack relies on metadata such as IP/AS endpoints, it cannot be run on link-layer encrypted networks (e.g., WPA2/WPA3). Nevertheless, a large fraction of public WiFi deployments remain unencrypted.

Training Phase. The adversary needs a dataset to compute endpoint commonness values, obtainable during the preliminary monitoring step. These values may become outdated over time [27], which may be addressed by periodic re-training.

Profiles. The limited number of profiles may understate difficulty in realistic settings with thousands of users, and profiles built from popular apps may not

match real ones. However, given millions of available apps, many users likely have a unique set of installed apps, which makes the attack feasible.

Tested Devices and OSes. Our setup uses only two Android smartphones and versions, but shared Android characteristics may generalize the attack to other Android devices. Future work can investigate whether iOS devices are vulnerable to similar attacks.

7.2 Countermeasures

Routing traffic via intermediate hops using VPNs and Tor can hide the endpoints observed at the access point. This may, however, increase latency and be vulnerable to more advanced traffic analysis attacks [25, 29]. A potential, but costly countermeasure could be to perturb profiles by sending cover traffic to additional endpoints, similar to defenses proposed against website fingerprinting attacks [16, 21].

8 Related Work

We review the state of the art in network traffic analysis, focusing on attacks on encrypted traffic and recent advances in device fingerprinting and tracking.

8.1 Traffic Analysis Attacks

Due to its security and privacy features, Tor is a prominent target of traffic analysis attacks [41, 45]. End-to-end confirmation attacks monitor the first and last hop of a circuit to match a client with a server [38, 39, 42]. Despite high reported success, the evaluation assumptions of end-to-end confirmation attacks may be unrealistic [46].

Website fingerprinting commonly train a ML model based on traffic features [43, 45, 48, 49, 12]. While website fingerprinting works even in the presence of link-layer encryption, many such attacks rely on unrealistic assumptions [27], such as their dependence on active user interactions, which must be modeled and simulated, and affects the validity of results. In contrast, our work uses passive background traffic and avoids reliance on active user interaction.

8.2 Deep Learning in Traffic Analysis

Deep learning has been applied to protocol detection [59], application identification from encrypted traffic [1], analysis of encrypted streaming data [47], and more practical settings such as app usage detection in cellular networks, where encrypted LTE/5G traffic still reveals application behavior [58]. An orthogonal line of research highlighted the limitations of these approaches, including generalization issues, dependency on large training datasets, and sensitivity to distribution shifts [8, 37, 14, 18]. In contrast, we show that effective fingerprinting is achievable with simple distance-based methods, without complex models.

8.3 Fingerprinting, Profiling, Tracking

Neumann et al. [40] compared histograms of 802.11 inter-arrival times of frames and Uluagac et al. [55] used clock skew on the wired backbone networks to classify devices. Both attacks relied on hardware or channel conditions, and they identified device models rather than users. Taylor et al. [52] identified the applications behind a flow from packet features, which required labelled training data for targets. Meneghello et al. [35] presented a passive traffic fingerprinting attack exploiting 5G/LTE control-channel information to identify smartphones and infer their location. Li et al. [33] proposed a graph learning framework for user profiling that, by modeling users, applications, locations, and time from spatiotemporal app usage data, inferred characteristics such as gender and age. Bahuguna et al. [10] introduced a deep-learning framework to identify mobile applications from encrypted traffic, and recognized user activities and user profiles. The WiFinger system used packet-level sequence matching to identify IoT events from noisy WiFi captures [31]. Todt et al. inferred user identities using channel characteristics broadcast by clients in WiFi 5 networks, namely beamforming feedback information [53]. Other researchers proposed methods to identify mobile devices despite modern protections such as MAC randomization, under realistic wireless network settings [36, 57]. Recent work also showed that devices and users remain distinguishable even under encryption and identifier obfuscation [26, 17].

While these works address device identification, event detection, or physical-layer leakage, our work links user traces across locations from passive background traffic. Our approach does not require active user interaction, explicit identifiers, or special hardware. Moreover, our attack does not rely on training compute-intensive ML models, which need to be periodically retrained due to cope with concept drift.

9 Conclusion

We introduced a passive background fingerprinting attack that uses background traffic generated automatically by mobile devices without requiring active behavior, which makes it more realistic compared to previous works. We show that even in the absence of payload access and persistent identifiers, metadata provide sufficient signals to distinguish user profiles and link traces across locations.

Our analysis identifies the main factors that influence performance, including endpoint representation, recording duration, and device characteristics. We show that autonomous system (AS) information, combined with transmission port, enables effective differentiation of user profiles, and longer observation periods improve the reliability of trace matching. Furthermore, the attack remains effective across different devices and application sets, highlighting its robustness. As a result, our findings show that passive background traffic alone can enable cross-location tracking of users, even when common privacy protections such as encryption and MAC randomization are in place. This highlights a gap in ex-

isting defenses and underlines the need for stronger protection mechanisms that address metadata leakage at the network level.

Data and Code Availability.

The code implementing the attack and analysis pipeline can be accessed at <https://github.com/shivaz22/fingerprinting-2026>. The captured traffic traces are not published by default, as raw packet captures may contain sensitive information. A sanitized version of the dataset is available on request from the corresponding author.

References

1. Aceto, G., Ciuonzo, D., Montieri, A., Pescapé, A.: Mobile encrypted traffic classification using deep learning. In: 2018 Network traffic measurement and analysis conference (TMA). pp. 1–8. IEEE (2018)
2. Agrawal, A., Bhatia, A., Bahuguna, A., Tiwari, K., Haribabu, K., Vishwakarma, D., Kaushik, R.: A survey on analyzing encrypted network traffic of mobile devices. *International Journal of Information Security* **21**(4), 873–915 (2022)
3. Android Developers: App startup time (2024), <https://developer.android.com/topic/performance/vitals/launch-time>, accessed: 2026-04-10
4. Android Developers: Optimize for doze and app standby (2024), <https://developer.android.com/training/monitoring-device-state/doze-standby>, accessed: 2026-04-10
5. Android Developers: Processes and app lifecycle (2024), <https://developer.android.com/guide/components/activities/process-lifecycle>, accessed: 2026-04-20
6. AppBrain Statistics: Most popular google play categories (2022), <https://www.appbrain.com/stats/android-market-app-categories>, accessed: 2022-12-20
7. Apple: Use private Wi-Fi addresses on Apple devices - Apple Support (Dec 2025), <https://support.apple.com/en-us/102509>, online; accessed 20. Apr. 2026
8. Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., Rieck, K.: Dos and don'ts of machine learning in computer security. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 3971–3988 (2022)
9. Auliya, S., Nugroho, L.E., Setiawan, N.A.: A review on smartphone usage data for user identification and user profiling. *Communications in Science and Technology* **6**(1), 25–34 (2021)
10. Bahuguna, A., Agrawal, A., Bhatia, A., Tiwari, K., Vishwakarma, D.: User profiling using smartphone network traffic analysis. In: 2021 International Conference on COMmunication Systems & NETworkS (COMSNETS). pp. 69–73. IEEE (2021)
11. Bar-Noy, A., Kessler, I.: Tracking mobile users in wireless communications networks. *IEEE Transactions on Information Theory* **39**(6), 1877–1886 (1993)
12. Bhat, S., Lu, D., Kwon, A., Devadas, S.: Var-cnn: A data-efficient website fingerprinting attack based on deep learning. arXiv preprint arXiv:1802.10215 (2018)

13. Bissias, G.D., Liberatore, M., Jensen, D., Levine, B.N.: Privacy vulnerabilities in encrypted http streams. In: International Workshop on Privacy Enhancing Technologies. pp. 1–11. Springer (2005)
14. Chatfield, K., Simonyan, K., Vedaldi, A., Zisserman, A.: Return of the devil in the details: Delving deep into convolutional nets. arXiv preprint arXiv:1405.3531 (2014)
15. Clarke, R., Wigan, M.: You are where you’ve been: The privacy implications of location and tracking technologies. *Journal of Location Based Services* **5**(3-4), 138–155 (2011)
16. Cui, W., Yu, J., Gong, Y., Chan-Tin, E.: Realistic cover traffic to mitigate website fingerprinting attacks. In: 2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS). pp. 1579–1584. IEEE (2018)
17. Cunha, M., Mendes, R., de Montjoye, Y.A., Vilela, J.P.: On the difficulty of not being unique: Fingerprinting users from wi-fi data in mobile devices. In: Proceedings of the 40th ACM/SIGAPP Symposium on Applied Computing. pp. 1335–1344 (2025)
18. Dacrema, M.F., Cremonesi, P., Jannach, D.: Are we really making much progress? A worrying analysis of recent neural recommendation approaches. In: Proceedings of the 13th ACM Conference on Recommender Systems. pp. 101–109 (2019)
19. Dingledine, R., Mathewson, N., Syverson, P.: Tor: The second generation onion router. In: *Usenix Security*. vol. 10. USENIX Association Berkeley, CA (2004)
20. Du, J., Xu, H., Tu, Z.: Personalized recommendation model of traffic package based on user consumption behavior. In: 2019 IEEE International Conference on Signal, Information and Data Processing (ICSIDP). pp. 1–6. IEEE (2019)
21. Gong, J., Wang, T.: Zero-delay lightweight defenses against website fingerprinting. In: 29th USENIX security symposium (USENIX security 20). pp. 717–734 (2020)
22. Group, T.T.: tcpdump and libpcap. <https://www.tcpdump.org/> (2024)
23. Gruteser, M., Liu, X.: Protecting privacy, in continuous location-tracking applications. *IEEE Security & Privacy* **2**(2), 28–34 (2004)
24. Hayes, J., Danezis, G.: k-fingerprinting: A robust scalable website fingerprinting technique. In: 25th USENIX Security Symposium (USENIX Security 16). pp. 1187–1203 (2016)
25. Heijligenberg, T., Lkhaoui, O., Kohls, K.: Leaky blinders: Information leakage in mobile vpns. In: International Conference on Applied Cryptography and Network Security. pp. 481–494. Springer (2022)
26. Huang, Y., Dong, Z., Yang, X., Zhang, D., Wang, Q., Wang, Z.: Smartphone user fingerprinting on wireless traffic. *IEEE Transactions on Mobile Computing* (2025)
27. Juarez, M., Afroz, S., Acar, G., Diaz, C., Greenstadt, R.: A critical evaluation of website fingerprinting attacks. In: Proceedings of the 2014 ACM SIGSAC conference on computer and communications security. pp. 263–274 (2014)
28. Kamar, M.E.Z.N., Esmailzadeh, A., Kim, Y., Taghva, K.: A survey on mobile malware detection methods using machine learning. In: 2022 IEEE 12th Annual Computing and Communication Workshop and Conference. pp. 0215–0221
29. Kohls, K., Rupprecht, D., Holz, T., Pöpper, C.: Lost traffic encryption: fingerprinting lte/4g traffic on layer two. In: Proceedings of the 12th Conference on Security and Privacy in Wireless and Mobile Networks. pp. 249–260 (2019)
30. Li, J., Zhou, H., Wu, S., Luo, X., Wang, T., Zhan, X., Ma, X.: FOAP: Fine-Grained Open-World Android App Fingerprinting. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 1579–1596 (2022)
31. Li, R., Liu, S., Hu, H., Ye, Q., Feamster, N.: Wifinger: Fingerprinting noisy iot event traffic using packet-level sequence matching (2025)

32. Li, T., Li, Y., Hoque, M.A., Xia, T., Tarkoma, S., Hui, P.: To what extent we repeat ourselves? discovering daily activity patterns across mobile app usage. *IEEE Transactions on Mobile Computing* **21**(4), 1492–1507 (2020)
33. Li, T., Li, Y., Zhang, M., Tarkoma, S., Hui, P.: You are how you use apps: user profiling based on spatiotemporal app usage behavior. *ACM Transactions on Intelligent Systems and Technology* **14**(4), 1–21 (2023)
34. Malinen, J.: Linux WPA/WPA2/WPA3/IEEE 802.1X Supplicant (2013), https://w1.fi/wpa_supplicant/
35. Meneghello, F., Rossi, M., Bui, N.: Smartphone identification via passive traffic fingerprinting: A sequence-to-sequence learning approach. *IEEE Network* **34**(2), 112–120 (2020)
36. Mishra, A.K., Péliissier, S., Cunche, M.: Fingerprinting connected wi-fi devices using per-network mac addresses. In: *International Symposium on Foundations and Practice of Security*. pp. 295–303. Springer (2024)
37. Musgrave, K., Belongie, S., Lim, S.N.: A metric learning reality check. In: *European Conference on Computer Vision*. pp. 681–699. Springer (2020)
38. Nasr, M., Bahramali, A., Houmansadr, A.: Deepcorr: Strong flow correlation attacks on tor using deep learning. In: *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. pp. 1962–1976 (2018)
39. Nasr, M., Houmansadr, A., Mazumdar, A.: Compressive traffic analysis: A new paradigm for scalable traffic analysis. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. pp. 2053–2069 (2017)
40. Neumann, C., Heen, O., Onno, S.: An empirical study of passive 802.11 device fingerprinting. In: *2012 32nd International Conference on Distributed Computing Systems Workshops*. pp. 593–602. IEEE (2012)
41. Oh, S.E., Sunkam, S., Hopper, N.: p1-fp: Extraction, classification, and prediction of website fingerprints with deep learning. *Proceedings on Privacy Enhancing Technologies* **2019**(3), 191–209 (2019)
42. Oh, S.E., Yang, T., Mathews, N., Holland, J.K., Rahman, M.S., Hopper, N., Wright, M.: Deepcoffea: Improved flow correlation attacks on tor via metric learning and amplification. In: *2022 IEEE Symposium on Security and Privacy (SP)*. pp. 1915–1932. IEEE (2022)
43. Panchenko, A., Lanze, F., Pennekamp, J., Engel, T., Zinnen, A., Henze, M., Wehrle, K.: Website fingerprinting at internet scale. In: *NDSS*. vol. 1, p. 23477 (2016)
44. Rahman, M.S., Sirinam, P., Mathews, N., Gangadhara, K.G., Wright, M.: Tiktok: The utility of packet timing in website fingerprinting attacks. *arXiv preprint arXiv:1902.06421* (2019)
45. Rimmer, V., Preuveneers, D., Juarez, M., Van Goethem, T., Joosen, W.: Automated website fingerprinting through deep learning (2017)
46. Rimmer, V., Schnitzler, T., Van Goethem, T., Romero, A.R., Joosen, W., Kohls, K.: Trace oddity: Methodologies for data-driven traffic analysis on tor. *Proceedings on Privacy Enhancing Technologies* (2022)
47. Schuster, R., Shmatikov, V., Tromer, E.: Beauty and the burst: Remote identification of encrypted video streams. In: *26th USENIX Security Symposium (USENIX Security 17)*. pp. 1357–1374 (2017)
48. Sirinam, P., Imani, M., Juarez, M., Wright, M.: Deep fingerprinting: Undermining website fingerprinting defenses with deep learning. In: *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. pp. 1928–1943 (2018)

49. Sirinam, P., Mathews, N., Rahman, M.S., Wright, M.: Triplet fingerprinting: More practical and portable website fingerprinting with n-shot learning. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security. pp. 1131–1148 (2019)
50. Sung, K., Biswas, J., Learned-Miller, E., Levine, B.N., Liberatore, M.: Server-side traffic analysis reveals mobile location information over the internet. *IEEE Transactions on Mobile Computing* **18**(6), 1407–1418 (2018)
51. Sánchez, P.M.S., Valero, J.M.J., Celdrán, A.H., Bovet, G., Pérez, M.G., Pérez, G.M.: A survey on device behavior fingerprinting: Data sources, techniques, application scenarios, and datasets. *IEEE Communications Surveys & Tutorials* **23**(2), 1048–1077 (2021)
52. Taylor, V.F., Spolaor, R., Conti, M., Martinovic, I.: Appscanner: Automatic fingerprinting of smartphone apps from encrypted network traffic. In: 2016 IEEE European Symposium on Security and Privacy (EuroS&P). pp. 439–454
53. Todt, J., Morsbach, F., Strufe, T.: Bfid: Identity inference attacks utilizing beam-forming feedback information. In: Proceedings of the 2025 ACM SIGSAC Conference on Computer and Communications Security. pp. 2399–2413 (2025)
54. Tukey, J.W., et al.: Exploratory data analysis, vol. 2. Addison-wesley Reading, MA (1977)
55. Uluagac, A.S., Radhakrishnan, S.V., Corbett, C., Baca, A., Beyah, R.: A passive technique for fingerprinting wireless devices with wired-side observations. In: 2013 IEEE Conference on Communications and Network Security (CNS). pp. 305–313
56. Velan, P., Čermák, M., Čeleda, P., Drašar, M.: A survey of methods for encrypted traffic classification and analysis. *International Journal of Network Management* **25**(5), 355–374 (2015)
57. Vogel, D., Viola, F., Kreimeyer, N.M., Meier, M.: I know you were here: Leveraging probe request templates for identifying wi-fi devices. In: 2024 20th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob). pp. 429–436. IEEE (2024)
58. Wang, J., Cheng, Z., Ordean, M., Cui, B.: What-app? app usage detection using encrypted lte/5g traffic. *Proceedings on Privacy Enhancing Technologies* (2026)
59. Wang, Z.: The applications of deep learning on traffic identification. *BlackHat USA* **24**(11), 1–10 (2015)
60. Wang, Z., Wei, G., Zhan, Y., Sun, Y.: Big data in telecommunication operators: data, platform and practices. *Journal of Communications and Information Networks* **2**(3), 78–91 (2017)
61. Wu, T., Xiao, X., Li, Q., Liu, Q., Hu, G., Luo, X., Jiang, Y.: Behavsniffer: Sniff user behaviors from the encrypted traffic by traffic burst graphs. In: 2023 20th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON). pp. 456–464. IEEE (2023)
62. Xu, Q., Zheng, R., Saad, W., Han, Z.: Device fingerprinting in wireless networks: Challenges and opportunities. *IEEE Communications Surveys & Tutorials* **18**(1), 94–104 (2015)

Reverse Engineering for Privacy-Preserving Validation of GWAS Outcomes

Yuzhou Jiang¹[0000–0001–9518–8103], Yuqiao Xu¹[0009–0009–3552–2136], and
Erman Ayday¹[0000–0003–3383–1081]

Case Western Reserve University, Cleveland, OH, USA
yxj466@case.edu, yxx914@case.edu, exa208@case.edu

Abstract. Reproducibility in genome-wide association studies (GWAS) is crucial for ensuring reliable genomic research outcomes. However, limited access to original genomic datasets (primarily due to privacy concerns) prevents researchers from reproducing experiments to validate results. In this paper, we propose a novel method for GWAS reproducibility validation that detects unintentional errors without requiring access to the original dataset. For each single nucleotide polymorphism (SNP), our approach reverse-engineers the GWAS contingency table for the case group using a grid search enhanced with several optimizations. We then compare the Minor Allele Frequency (MAF) derived from the reconstructed contingency table row to the MAF reported in the research findings and evaluate the difference. If the difference falls within a predefined threshold for all SNPs, the results are considered valid. Otherwise, the study, especially the SNPs that fail validation, is flagged for further inspection. We evaluate our approach using three real-world and additional synthetic datasets generated by simulating ground truth contingency tables, and demonstrate that our scheme can efficiently detect even very subtle errors. This novel validation technique offers a promising solution to the GWAS reproducibility challenge.

Keywords: Genomic privacy, reproducibility, genome-wide association studies

Introduction

Genome-wide association studies (GWAS) enable scientists to uncover associations between genetic variants and traits or diseases [3, 12, 10, 1]. These studies offer invaluable insights by examining large-scale genomic datasets to detect associations between specific single nucleotide polymorphisms (SNPs) and various phenotypic traits. As a result, GWAS has not only led to numerous groundbreaking discoveries, but it also serves as a crucial foundation for future research, guiding subsequent studies and advancing our understanding of complex genetic relationships. The increasing reliance on GWAS findings introduces a critical challenge: **the issue of reproducibility**. In the context of GWAS, reproducibility involves using the original genomic dataset and applying the same analytical approach to verify the reported outcomes [2, 14]. Reproducibility

also plays a critical role in securing funding for GWAS projects [11]. Independent replication of research outcomes is often an expectation from grant reviewers, as it is a key indicator of scientific rigor and reliability. Without the ability to validate findings through reproducibility, securing grants for GWAS research becomes increasingly difficult.

In this paper, we focus on Single Nucleotide Polymorphisms (SNPs) and propose a novel scheme, REGenOV (*Reverse Engineering for Genomic Research Outcome Validation*), to enable the validation of GWAS outcomes. Unlike existing approaches [5, 9, 7], our approach does not require sharing the target genomic dataset in any form. Instead, it relies solely on publicly available data and summary statistics provided by the researcher. We evaluate the trustworthiness of research findings on a per-SNP basis by reconstructing the contingency table for each SNP and verifying the consistency of the Minor Allele Frequency (MAF) in the case group.

For each SNP, we first reconstruct the control row of the contingency table using publicly available data since it is common to identify publicly available datasets that are representative of the control population in a GWAS. Next, we perform a grid search, incorporating several optimizations, to reverse-engineer the potential case row(s) that may reproduce the reported GWAS experiment statistic, such as chi-square or odds ratio. We then calculate the Minor Allele Frequency (MAF) based on the inferred case row(s) and compare it with the MAF reported in the research findings. If the minimum difference between the two MAFs falls within the resolution used in the reported GWAS statistic for all SNPs in the findings, the verifier can trust the reported results and incorporate them into their research process. Otherwise, the verifier may consider the GWAS outcome for some SNPs to be potentially erroneous and may conduct further investigation, such as requesting additional details, pending Institutional Review Board (IRB) approval.

We evaluate our proposed scheme on three genomic datasets from the OpenSNP project [15], as well as on synthetic datasets of larger sizes, with respect to detecting errors in GWAS outcomes and time complexity. Throughout the experiments, our approach demonstrates robustness and it consistently identifies errors in real-world scenarios without requiring additional information from the original research. This validates its effectiveness and practicality for reproducibility verification in genomic studies.

Related Work

Verifying GWAS outcomes typically requires access to the original dataset, but privacy concerns prevent researchers from sharing sensitive genomic data due to its vulnerability to reidentification and membership inference risks [8, 16, 4, 6]. Sankararaman et al.[16] demonstrated that the summary statistics of a target group can be used to determine whether an individual is present in a dataset within a study cohort. Deznabi et al.[4] showed that sample-wise correlations (e.g., kinship) can be used to infer hidden parts of genomic data by leveraging familial information

from public sources. Gymrek et al. [6] further revealed that an individual’s surname can be recovered using Y chromosome data. Due to these vulnerabilities, researchers are often unwilling to share even summary statistics of the target group. Without access to such data, independent researchers cannot replicate the study or validate its outcomes, making reproducibility checks impractical.

Existing privacy-preserving techniques [5, 9, 7] for reproducibility rely on sharing additional, privacy-protected information from researchers. Differential privacy, introduced by Dwork [5], is one of the most popular approaches for protecting sensitive data under a quantifiable privacy guarantee, and it has been applied to genomic privacy in GWAS validation. Halimi et al.[7] proposed a scheme for detecting GWAS errors by comparing reported GWAS results with those recalculated from a noisy version of the private dataset. However, their method struggles to identify a reliable threshold that gives high confidence in determining whether a given SNP is correct. Jiang et al.[9] proposed sharing a differentially private version of the dataset to enable verifier-side GWAS outcome validation. Although this approach demonstrates high sensitivity to small errors, it still lacks a well-defined threshold for determining correctness across datasets. These limitations underscore the need for a reproducibility validation method that avoids sharing any additional data while still enabling effective and reliable validation.

System Setting

We consider two main entities: **Alice**, a researcher who conducts genome-wide association studies (GWAS) and publishes the findings, and **Bob**, another researcher who seeks to validate these GWAS outcomes.

Alice (Researcher) conducts GWAS on a local genomic dataset to identify associations between specific SNPs and phenotypic traits or diseases. Her research findings typically include (1) the number of samples in the case and control groups and, for each SNP showing significant association with a trait or disease, (2) its rsID, (3) the statistical value (e.g., χ^2) and p-value from the GWAS experiments, and (4) metadata, such as the minor allele frequency (MAF) of the case group.

We assume that Alice uses a publicly available reference dataset as the control group. If the control dataset is not publicly available, she provides summary statistics sufficient to construct the control row of the contingency table for each SNP (a required step in the proposed scheme). This sharing complies with NIH genomic privacy policies [13] for the following two reasons. First, it minimizes the risk of re-identification by utilizing only aggregate statistics rather than individual-level data, which is permitted under the policy. Second, since the reference dataset does not correspond to the examined phenotype/disease, it does not pose a privacy risk for inference attacks on the case group. Therefore, our proposed setting ensures compliance with privacy regulations while enabling validation.

Alice’s goal is to enable external validation of her findings while ensuring the privacy of the genomic dataset. We assume she is willing to provide

any information that is permissible under NIH genomic privacy policies [13] to facilitate validation. However, she does not intend to share the genomic dataset used in her research under any circumstances, regardless of whether privacy-preserving technologies are applied to it. Alice also understands that existing privacy-preserving data-sharing schemes either pose privacy risks, introduce excessive noise that compromises GWAS validation, or suffer from impractical implementation constraints.

Bob (Verifier) is another researcher interested in validating the GWAS results published by Alice. Bob has access to 1) all the information included in Alice’s research findings and 2) the public reference dataset used by Alice (or, if the dataset is private, the summary statistics she provides). His goal is to verify whether the GWAS statistic for each SNP can be trusted based on the given information.

In our framework, we assume that Alice, the researcher, is honest but may unintentionally introduce errors during the GWAS process. Computational mistakes or errors in reporting research findings could lead to incorrect GWAS outcomes, potentially misleading other researchers. Our approach is designed to detect such unintentional errors and flag the findings for further investigation if inconsistencies arise during the validation process.

We also consider the possibility of multiple independent errors occurring throughout the GWAS process. These errors may arise at different stages, such as during data collection, preprocessing, or statistical analysis, but are treated as separate events. Importantly, our approach remains effective in detecting such compounded errors. This is because multiple independent errors are unlikely to align in a way that produces results identical to those of a correctly executed GWAS. In other words, the probability that a series of independent mistakes would lead to the same outcome as a valid GWAS is extremely low, making discrepancies easier to identify.

We do not account for scenarios where Alice deliberately generates incorrect results and publishes them as research findings. This could involve fabricating datasets, intentionally modifying values within the dataset, or altering GWAS experiment results. Such misconduct, if discovered, would have severe consequences for the researcher, including loss of reputation, reduced funding opportunities, damaged collaborations, and exclusion from the genomics research community. However, detecting such actions requires direct access to the original dataset obtained from real patients or individuals, which is unlikely for third parties. This limitation reflects a broader challenge in data validation, and no existing methods can fully resolve it.

As previously discussed, the proposed mechanism can also be used by Alice as a self-validation method, allowing her to verify her own GWAS findings before publication. However, in this paper, we focus on the researcher-verifier setting while introducing our scheme. The self-validation scenario is illustrated later in the **Discussion** section.

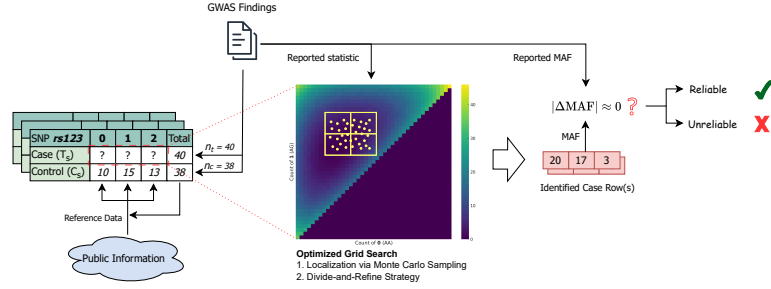


Fig. 1: Bob’s SNP-level workflow for validating GWAS outcomes. The process is applied to each reported SNP, and the final validation decision is made by aggregating evidence across all significant SNPs.

1. **Contingency-table construction.** For each SNP i in the reported findings, Bob constructs a contingency table in which the control row C_s^i and sample sizes are prefilled using public information and Alice’s reported GWAS results.
2. **Case-row reconstruction.** Bob performs a grid search using two optimization strategies: localization via Monte Carlo sampling and divide-and-refine. He then identifies the candidate case row or rows that may generate the target statistic.
3. **MAF consistency check.** Bob computes the difference between the minor allele frequency (MAF) derived from the reconstructed case row and the reported MAF.
4. **Aggregate validation.** Bob aggregates MAF differences across all SNPs. If the candidate rows satisfy $\Delta_{\text{MAF}} \approx 0$ across SNPs, the findings are considered reliable. Otherwise, deviations may trigger clarification or further institutional review.

Methodology

In this section, we introduce our proposed scheme for enabling the validation of GWAS outcomes: **REGenOV** (*Reverse Engineering for Genomic Research Outcome Validation*). REGenOV performs validation on the entire set of genomic findings at the SNP level, evaluating each reported SNP individually and aggregating the results. If an inconsistency is detected for a particular SNP, it may indicate potential errors in the GWAS findings, prompting further verification or additional investigation. Figure 1 illustrates this validation workflow.

The section is structured as follows. First, we provide details of REGenOV with the baseline brute-force grid search method and discuss its limitations. Then, we introduce optimizations, including localization via Monte Carlo sampling and a divide-and-refine strategy.

For clarity, we use the chi-square (χ^2) statistic as an example of the GWAS experiment throughout this section. However, REGenOV is also applicable to other statistical tests, e.g., the odds ratio, which we examine in the **Evaluation** section.

Baseline: Brute-Force Reverse Engineering for GWAS Outcome Validation. REGenOV consists of two key stages: contingency table reconstruction and validation through MAF comparison. We provide detailed explanations of these stages as follows.

Stage 1: Contingency Table Reconstruction. In this stage, REGenOV reconstructs the case row of the contingency table for each SNP using a publicly available reference dataset used in the research findings. If the reference dataset is not available to the verifier, REGenOV reconstructs the control row using the summary statistics provided by Alice. The challenge then is to infer the case row that, when combined with the control row, reproduces the published chi-square (χ^2) statistic. For a given SNP i , we represent the contingency table as

$$\mathbf{CT}^i = \{T_s^i, C_s^i\}, \quad (1)$$

where $T_s^i = \{T_0^i, T_1^i, T_2^i\}$ represents case group counts (i.e., the case row) and $C_s^i = \{C_0^i, C_1^i, C_2^i\}$ represents the control group counts (i.e., the control row). The number of samples in the case group is denoted as n_t , and the number of samples in the control group is denoted as n_c . The objective is to identify a valid configuration of T_s^i such that the reported chi-square statistic (research outcome provided by the researcher) satisfies:

$$\chi_{\text{obs}}^2(\mathbf{T}_s^i) \approx \chi_{\text{reported}}^2, \quad (2)$$

where $\chi_{\text{obs}}^2(\mathbf{T}_s^i)$ represents the chi-square statistic computed from the reconstructed case row $\mathbf{T}_s^i$.

Our initial algorithm for contingency table reconstruction is a brute-force grid search. The grid search exhaustively enumerates all possible distributions of the case group's genotype counts. This is mathematically equivalent to distributing n_t elements into three categories (e.g., $\mathbf{AA}$, $\mathbf{AG}$, $\mathbf{GG}$), leading to the number of possible genotype configurations equal to $\binom{n_t+2}{2}$. Among these candidates, we identify the case row(s) whose observed chi-square statistic $\chi_{\text{obs}}^2(T_s^{i'})$ is equal to the reported chi-square value χ_{reported}^2 from the research findings.

If multiple candidate case rows satisfy this condition, rather than arbitrarily selecting one, we pass all qualifying candidates to the second validation stage for further assessment.

Stage 2: Validation Through MAF Comparison. Once we have identified the candidate case rows for a given SNP, the next step is validation using the minor allele frequency (MAF) reported in the research findings. The core idea behind this validation approach is that there is a strong relationship between MAF and the contingency table of a SNP. If the GWAS process is error-free, both the chi-square statistic and the MAF should match the values reported in the research findings.

While it is true that a match in both chi-square and MAF does not *strictly* guarantee that the GWAS calculation was performed correctly (since chi-square and MAF are not in a one-to-one mathematical relationship), this is not a concern in an honest setting. Under normal conditions, even if Alice unintentionally makes an incorrect GWAS calculation or reporting error, it is highly unlikely that such an error would coincidentally result in a case row that matches both the chi-square statistic and the MAF value.

Thus, if a reconstructed case row 1) yields a chi-square statistic that matches or closely approximates the reported value and 2) simultaneously produces an MAF value consistent with the published findings, we can have high confidence that the GWAS analysis was conducted correctly. This enables our validation scheme to effectively detect inconsistencies by focusing on MAF comparison.

If multiple candidate case rows satisfy the chi-square constraint for a SNP, we compute the MAF for each candidate and take the minimum absolute difference between the reconstructed and reported MAF values:

$$\Delta_{\text{MAF}} = \min |\text{MAF}_{\text{case}}^{i'} - \text{MAF}_{\text{reported}}^i|. \quad (3)$$

Our validation approach is designed to achieve high recall, meaning it minimizes false negatives (Type II errors). If we can find one candidate case row with $\Delta_{\text{MAF}} \approx 0$ for all SNPs reported to have significant associations with the disease or trait of interest, we consider the entire GWAS experiment to be valid. Otherwise, if one or more SNPs have MAF values that deviate from the reported ones, Bob may doubt the correctness of the reported results. He may request further clarification, additional data, or initiate a formal investigation through the Institutional Review Board (IRB).

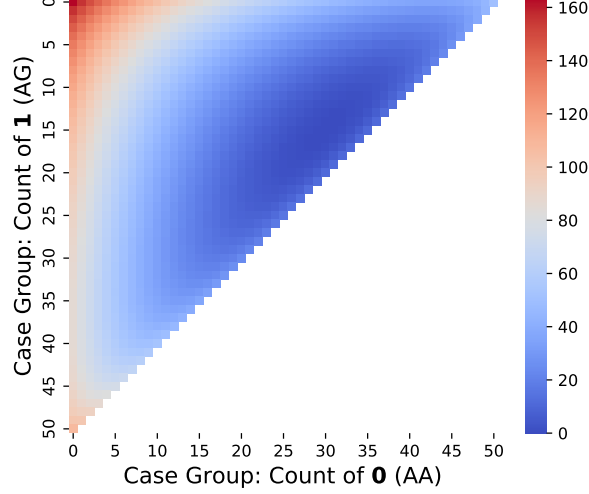
Limitations of Brute-Force Grid Search. Despite its simplicity, brute-force grid search suffers from significant computational inefficiencies. For a total of N case samples, an exhaustive search requires evaluating all possible genotype distributions, leading to a time complexity of $O(N^2)$ for each reported SNP. While this approach may be feasible for small genomic datasets with thousands of case samples, it becomes computationally prohibitive for larger datasets, making it impractical for large-scale GWAS validation.

Improving the Grid Search Strategy. We analyze the chi-square heatmap, where the x-axis represents the number of homozygous major genotypes (e.g., **AA**) in the case group and the y-axis represents the number of heterozygous genotypes (e.g., **AG**) in the case group, as shown in Figure 2. This 2D representation is possible because the count of homozygous minor genotypes (e.g., **GG**) can be determined as $T_2 = n_{\text{case}} - T_0 - T_1$. Based on this observation and preliminary experiments, we propose two optimization strategies to improve the search process: divide-and-refine and localization via Monte Carlo sampling.

1. Divide-and-Refine. As illustrated in Figure 2, the chi-square statistic exhibits a generally concave pattern across the search space, with some minor local fluctuations. While these fluctuations are not easily distinguishable from the heatmap itself, we confirmed their presence through finer-grained sampling in our preliminary experiments.

We take advantage of this feature by proposing a divide-and-refine strategy to efficiently locate genotype configurations that match the reported chi-square statistic. We begin with an initial bounding box that serves as the search space and recursively divide it into four equally sized (or nearly equal) sub-rectangles.

For each rectangle, we evaluate the chi-square statistic at its four vertices. If all four vertices fall within the range $[\chi_{\text{target}}^2 - d, \chi_{\text{target}}^2 + d]$, where d

Fig. 2: A χ^2 heatmap.

is a relaxation parameter to account for local fluctuations, we perform an exhaustive grid search within the rectangle since all internal points may be valid solutions. Otherwise, we determine whether to prune the region or subdivide further. We refine the search deeper if either of the following conditions holds:

1. **At least one vertex corresponds to an invalid genotype combination (i.e., $T_2 < 0$).** This indicates that the rectangle contains both valid and invalid regions, making it difficult to analyze without further subdivision.
2. **At least one vertex satisfies $\chi^2 \geq \chi_{\text{target}}^2 - d$ and at least one satisfies $\chi^2 \leq \chi_{\text{target}}^2 + d$.** This suggests that the region may contain a valid solution under the relaxed condition.

With this optimization, the time complexity of the proposed REGenOV is given by:

$$O(N \log N + fN^2), f \ll 1, \quad (4)$$

where $O(N \log N)$ corresponds to the divide-and-refine step, which follows a quadtree-like recursive subdivision, and $O(fN^2)$ accounts for the final grid search within the pruned space, where f represents the fraction of the total search space satisfying the chi-square threshold condition $[\chi_{\text{target}}^2 - d, \chi_{\text{target}}^2 + d]$.

2. Localization via Monte Carlo Sampling. Lower chi-square statistic values, which correspond to lower p-values and thus stronger associations with traits and diseases, are concentrated within a relatively small region of the entire space (i.e., the dark blue regions in Figure 2). This observation raises the question: *Can we leverage this structure to further prune the search space?*

To efficiently identify the search space, we employ a Monte Carlo sampling approach. Instead of searching the entire solution space exhaustively, we first localize the region of interest and then perform a fine-grained search within that subregion. We generate M random samples of valid case-group genotype pairs (T_0, T_1) and evaluate their chi-square statistics. Using these sampled points, we determine the localized search space by selecting all points where:

$$\chi_{\text{obs}}^2(T_0, T_1) \leq \chi_{\text{target}}^2 + d. \quad (5)$$

Here, d serves as an upper threshold to accommodate minor fluctuations. It is shared with the divide-and-refine strategy to ensure consistency across both optimization methods.

Once the sampled points satisfying the chi-square constraint are identified, we compute the bounding box enclosing the region of interest with a relaxation factor t :

$$B_t = [\min T_0 - t, \max T_0 + t, \min T_1 - t, \max T_1 + t]. \quad (6)$$

This adjustment ensures that small perturbations in chi-square values do not inadvertently exclude valid solutions, making the localization process more robust. The expansion by t helps capture edge cases where sharp chi-square transitions near significant association zones (e.g., the darkest blue regions along the diagonal in Figure 2) may otherwise lead to local fluctuations.

The time complexity of this optimization is $O(M)$, where M is the number of Monte Carlo sampling trials.

Overall Time Complexity. After the aforementioned optimizations, the overall time complexity of REGenOV is:

$$O(N \log N + fN^2 + M), \quad (7)$$

where N denotes the number of case samples, M is the number of Monte Carlo sampling trials, and f represents the fraction of the actual search space that satisfies the relaxed chi-square threshold boundaries. Additionally, we evaluate the time efficiency of our approach experimentally in the next section.

Evaluation

Dataset. We use three real-world datasets with different phenotypes: lactose intolerance, hair color, and eye color, obtained from the OpenSNP project [15]. The lactose intolerance dataset contains 9,091 SNPs from 60 individuals. The hair color dataset includes 9,686 SNPs, also from 60 individuals. The eye color dataset consists of 28,396 SNPs from 401 individuals.

Additionally, we simulate the validation process on large-scale datasets by generating synthetic contingency tables. For each trial, we randomly generate a case row constrained by the number of case samples n_t , and a control row of size $n_c = n_t$. We then compute the chi-square (χ^2) statistic

from the synthetic table and assume that both the control row and the χ^2 statistic are available to the verifier, consistent with our system setting. This synthetic setup allows us to evaluate the performance and scalability of REGenOV under realistic, large-scale genomic workloads.

Modeling GWAS Errors. We model GWAS errors in two primary ways. In the first case, we assume that unintentional calculation errors may occur, leading to inaccuracies in the GWAS results for a SNP. We model such errors by introducing Gaussian noise to the GWAS experiment outcome and reporting $\chi_{\text{true}}^2 + \mathcal{N}(0, \beta_d)$ as the result. We refer to this type of error as **deviation error**. A larger β_d corresponds to a greater distortion caused by computational errors. To ensure validity, we truncate any negative values to zero.

The second error occurs when researchers unintentionally report an incorrect SNP ID while collecting and summarizing results. In this scenario, either the reported statistical summaries do not correspond to the actual SNP, or the MAF and GWAS experiment results are mismatched, or all reported values are incorrect. We model this error by randomly sampling a GWAS experiment result within the range $[\chi_{\text{true}}^2 \times (1 - \beta_r), \chi_{\text{true}}^2 \times (1 + \beta_r)]$. This error is referred to as **reporting error**. The parameter β_r determines the degree of distortion introduced in the reported statistical values, where a larger β_r results in greater deviations from the true values.

It is important to note that these errors are unintentional and independent of each other, making the simultaneous occurrence of both errors unlikely. Even if both errors were to occur, it would be highly improbable for the combination of errors to produce the same GWAS experiment results as the original, error-free findings. As a result, we do not model scenarios where both errors occur together, as their combined effect would still lead to significant deviations. These deviations would make it unlikely for the results to match the publicly available MAFs. If a mismatch in MAF is detected, the GWAS findings should be reconsidered and further investigated. For simplicity, we model these two types of errors separately.

Experiment Setting. We use detection accuracy as our primary evaluation metric, defined as the proportion of SNPs with injected errors that are correctly identified as erroneous. Since validation in our scheme is performed SNP-by-SNP, this metric reflects the accuracy of detecting inconsistencies at the SNP level. Since REGenOV validates each SNP individually and aggregates the results, the detection of a single erroneous SNP implies that the overall GWAS findings would be flagged for further scrutiny. Thus, detection accuracy also serves as a proxy for assessing the correctness of the entire GWAS experiment.

If not specified otherwise, we use the following parameter configuration. We assume that Alice employs the χ^2 test in her GWAS analysis. For each dataset, we construct a public reference dataset that does not include the corresponding phenotype and use it to generate the statistical summaries that Alice would have utilized in her GWAS. We set the significance threshold to $\alpha = 0.05$ for three smaller datasets (lactose intolerance, hair color, and hand color), meaning that all SNPs with a p-value below

0.05 are considered significantly associated with the phenotype and are included in the final GWAS findings. For the eye color dataset, we set $\alpha = 0.005$ to account for the larger sample size. To ensure numerical stability and avoid any zero-count issues, we apply a pseudo-count of 0.5 to all genotype counts.

For our scheme, we set the number of Monte Carlo sampling iterations to $M = \max(1000, n_{\text{case}})$, ensuring that the number of samples is sufficient for rough localization. We set the relaxation parameters to $d = \max(5, 0.1 \times \chi_{\text{reported}}^2)$ and $t = 5$, allowing flexibility in capturing local fluctuations in the test statistic. The resolution tolerance is set to 10^{-6} to match the precision level of Alice’s GWAS experiment results.

Regarding error modeling, we vary the error parameters β_r and β_d from 10^{-6} to 10^{-1} . To ensure reliability, all experiments are repeated 100 times, and we report the average results.

GWAS Outcome Validation Against Errors. We begin by evaluating the accuracy of GWAS validation on three real-world datasets in the absence of errors. Our experimental results show that the proposed scheme consistently identifies 100% of correctly reported SNPs as reliable.

Next, we assess its performance in detecting two types of errors, with the results presented in Figures 3a and 3b. The y-axis represents the detection accuracy of errors, where a higher percentage indicates better performance. Our scheme achieves nearly 100% detection accuracy when the error parameters satisfy $\beta_r \geq 10^{-4}$ for reporting errors and $\beta_d \geq 10^{-3}$ for deviation errors, both of which are subtle. These results demonstrate the effectiveness of our approach, highlighting its high sensitivity to errors.

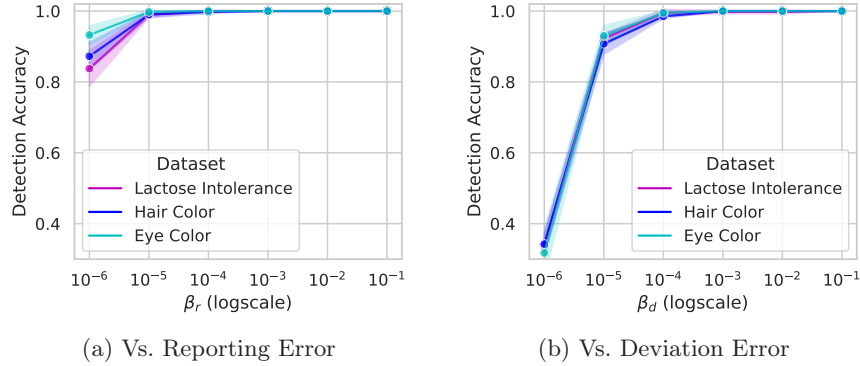


Fig. 3: Performance of GWAS outcome validation in detecting errors in χ^2 tests across three datasets.

On Synthetic Datasets with Larger Dimensions. We further evaluate our scheme on synthetic genomic datasets of larger sizes to ensure its scalability. For each case sample size in 100, 1,000, 10,000, 100,000,

1,000,000, we perform 20 trials per dataset size under each parameter configuration. We observe similar results as with the real-world datasets in Figures 3a and 3b. This confirms that our scheme is applicable to larger datasets.

One important observation is that, on larger datasets ($n_t \geq 1000$), we typically do not need to find all possible case rows, which would be computationally expensive, as the test statistic is less likely to take the same value across different case rows. Instead, we can terminate early once an eligible case row is found. In this case, if discrepancies in MAF are detected, we should repeat the validation using a “non-pruning run” to further assess whether the reported result is trustworthy.

Alternative GWAS Experiment (Odds Ratio). Similar to the χ^2 test, we observe high detectability against small errors, as shown in Figure 4. This confirms that our scheme can also be applied to other types of GWAS experiments. To utilize our validation scheme, a GWAS experiment should at least satisfy the following two conditions:

1. It must be computable solely based on the contingency table.
2. It should exhibit a roughly concave shape, allowing for some minor local fluctuations.

Impact of Precision in Reported Statistics. Alice may not always share highly accurate p-values with the GWAS findings. In some cases, the statistic associated with the reported SNPs may be pruned, potentially affecting validation performance. We conducted experiments to assess how varying precision impacts our scheme’s performance. Figure 5 presents the results of the lactose intolerance dataset. We observe that high precision in p-value reporting ensures the most reliable validation. For all resolution digits ≥ 4 against reporting errors, our scheme can detect errors under β_r and β_d at the 10^{-3} level, whereas resolutions ≤ 3 exhibit lower confidence in error detection. For deviation errors, at least five decimal places are required to effectively counter a 10^{-3} error. These findings demonstrate that while our scheme remains capable of detecting small errors, it is always recommended to use a higher resolution in research findings for optimal reliability.

Time Efficiency . We present the running time results for a single SNP on datasets with varying numbers of case samples in Table 1. Each configuration was tested using 20 independent trials to ensure consistency, and all experiments were performed using a single thread. The benchmarks were conducted on a 5900X 3.7 GHz processor with 32 GB of memory. The number of control samples was set equal to the number of case samples, with the control group genotype distribution [AA, AG, GG] fixed at a 6:3:1 ratio. Case rows were randomly sampled to produce significant associations in the GWAS test. As shown in Table 1, the execution time increases approximately linearly with dataset size. Our scheme successfully validates datasets with up to 1 million case samples, demonstrating its high scalability and efficiency under realistic conditions.

Validator Robustness to Reporting Errors

We evaluate how reliably the validator distinguishes correctly reported GWAS statistics from corrupted reports. For each phenotype—lactose

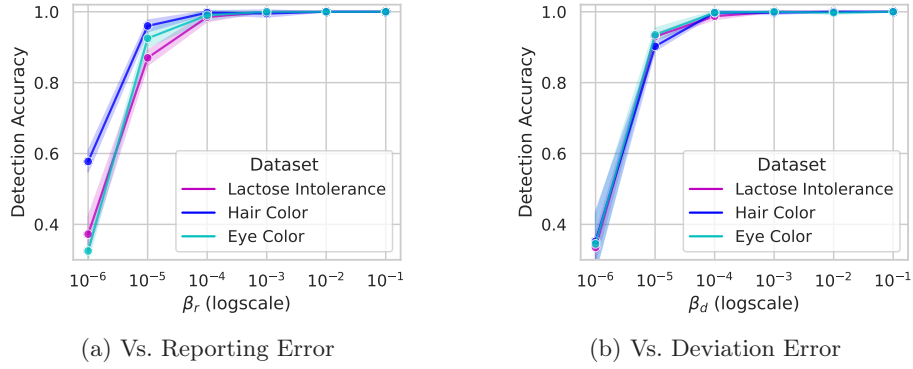


Fig. 4: Performance of GWAS outcome validation on detecting errors in odds ratio tests across three datasets.

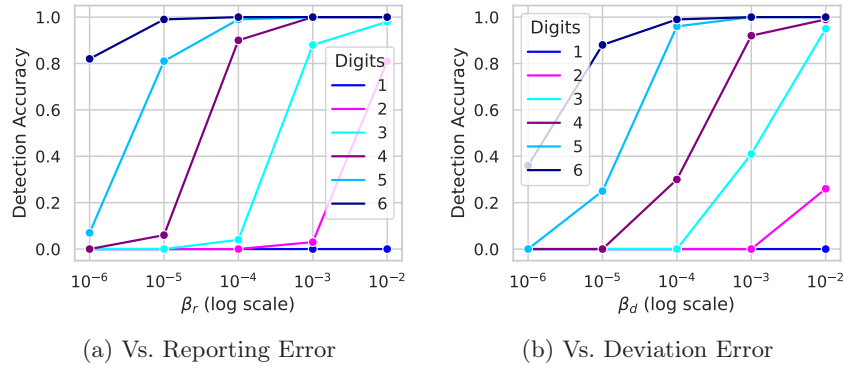


Fig. 5: Impact of statistic precision on validation accuracy.

intolerance, hair color, and eye color—we repeatedly sample SNPs, construct case-control genotype count rows, and recompute the reference Pearson χ^2 statistic for each sampled SNP, following the methodology described earlier. Paired valid/corrupted trials are generated per SNP as described in the evaluation setup for GWAS outcome validation against reporting errors.

In this experiment, the validator’s decision rule uses only the reported statistical test value. A report is accepted if the reported Pearson χ^2 statistic matches the recomputed reference statistic within a fixed tolerance δ . We intentionally do not use Minor Allele Frequency (MAF) consistency as an additional condition, so the experiment measures whether statistical test values alone are sufficient to distinguish valid from corrupted reports. Operating-point estimates use $\delta = 10^{-4}$ with $N = 1000$ paired trials per dataset (seed = 0). ROC curves are produced by sweeping δ over a logarithmic grid.

Table 1: Execution time across varying numbers of case samples.

n_t	100	1,000	10,000	100,000	1,000,000
t (s)	0.5746	6.8334	171.2239	1348.3858	7565.7778

Table 2: Validator performance at $\delta = 10^{-4}$.

Dataset	TPR	FNR	FPR (95% CI)	AUC
Lactose	1.000	0.000	0.201 (0.177–0.227)	0.965
Hair	1.000	0.000	0.304 (0.276–0.333)	0.832
Eye	1.000	0.000	0.226 (0.201–0.253)	0.998

Across all three datasets, the validator achieves perfect sensitivity, with $\text{TPR} = 1.0$ and $\text{FNR} = 0.0$. Because the 95% confidence intervals for FNR have an upper bound of approximately 0.0038 in all cases, Table 2 reports only FPR confidence intervals. False positive rates vary by phenotype: 0.201 for lactose intolerance, 0.304 for hair color, and 0.226 for eye color. ROC analysis confirms strong discrimination, with AUC values of 0.965, 0.832, and 0.998, respectively.

Discussion

Individual SNP GWAS Outcome Validation. REGenOV can also be used as a single-SNP validation tool when the verifier is interested in assessing the trustworthiness of a specific SNP result rather than validating the entire set of reported findings. In this scenario, the verifier runs the validation procedure only for the SNP of interest and checks whether the reconstructed candidate case row produces an MAF value consistent with the reported MAF. If the MAF difference is within the reporting tolerance, the SNP result can be treated as internally consistent with the published statistic and metadata. Otherwise, the result may be considered potentially unreliable, and the verifier may request clarification, additional documentation, or further institutional review as described in the system setting.

Incremental Privacy Disclosure. REGenOV does not require Alice to disclose individual-level genotypes, raw case-group data, or a privacy-preserving version of the target dataset. The verifier operates only on information that is already part of the reported GWAS findings, together with public reference data or aggregate control statistics. However, the validation procedure may infer one or more candidate aggregate case rows that are consistent with the reported statistic. Therefore, the privacy claim of REGenOV is not that validation reveals no information, but rather that it enables reproducibility checking without requiring any additional raw-data sharing or individual-level access to the target genomic dataset. The inferred rows are aggregate SNP-level candidates and may not be unique, but they should still be understood as part of the information revealed by the validation process.

Self-Verification as a Quality Control Step. As discussed earlier, REGenOV can also be used by Alice as a self-checking mechanism before publication. Although our main setting focuses on an external researcher-verifier, the same validation logic can help researchers identify reporting or computation errors in their own GWAS findings before the results are released.

Specifically, Alice can perform local validation by applying the same steps as Bob to her prepared GWAS findings. She inputs the reported GWAS statistics, reference data or control summaries, and the sample sizes of the case and control groups. REGenOV then reconstructs candidate case rows and compares the MAFs implied by those rows with the MAF values included in the prepared findings. If inconsistencies are detected, Alice can inspect the corresponding SNPs before publication.

This self-validation use case does not replace independent replication or external review, but it provides a lightweight quality-control step for detecting unintentional errors. By completing this check before publication, Alice can provide additional assurance that the reported GWAS statistics and metadata are internally consistent, thereby strengthening the credibility of the released findings.

Conclusion

In this paper, we proposed *REGenOV*, a novel method for validating GWAS findings without requiring access to sensitive genomic data. Our approach relies solely on researcher-provided summary statistics and publicly available reference data. By reverse-engineering the case rows of contingency tables for each SNP and verifying the consistency of Minor Allele Frequencies (MAFs), REGenOV enables accurate SNP-wise validation of GWAS outcomes. Our method effectively detects unintentional errors, remains robust under reduced statistical precision, and supports self-validation by researchers prior to publication. REGenOV provides a practical, privacy-preserving solution to enhance the trust and reproducibility of genomic research. Future work may explore extending REGenOV to additional GWAS models or multi-cohort validation.

Acknowledgement

The work was partly supported by the National Science Foundation (NSF) under grant number 2141622 and the National Institutes of Health (NIH) under grant number R01LM014520.

References

1. Abdel Abdellaoui, Loic Yengo, Karin JH Verweij, and Peter M Visscher. 15 years of gwas discovery: realizing the promise. *The American Journal of Human Genetics*, 110(2):179–194, 2023.
2. C Glenn Begley and John PA Ioannidis. Reproducibility in science: improving the standard for basic and preclinical research. *Circulation research*, 116(1):116–126, 2015.
3. Rita M Cantor, Kenneth Lange, and Janet S Sinsheimer. Prioritizing gwas results: a review of statistical methods and recommendations for their application. *The American Journal of Human Genetics*, 86(1):6–22, 2010.
4. Iman Deznabi, Mohammad Mobayen, Nazanin Jafari, Oznur Tantan, and Erman Ayday. An inference attack on genomic data using kinship, complex correlations, and phenotype information. *IEEE/ACM transactions on computational biology and bioinformatics*, 15(4):1333–1343, 2017.
5. Cynthia Dwork. Differential privacy. In *International colloquium on automata, languages, and programming*, pages 1–12. Springer, 2006.
6. Melissa Gymrek, Amy L McGuire, David Golan, Eran Halperin, and Yaniv Erlich. Identifying personal genomes by surname inference. *Science*, 339(6117):321–324, 2013.
7. Anisa Halimi, Leonard Dervishi, Erman Ayday, Apostolos Pyrgelis, Juan Ramon Troncoso-Pastoriza, Jean-Pierre Hubaux, Xiaoqian Jiang, and Jaideep Vaidya. Privacy-preserving and efficient verification of the outcome in genome-wide association studies. *arXiv preprint arXiv:2101.08879*, 2021.
8. Nils Homer, Szabolcs Szelinger, Margot Redman, David Duggan, Waibhav Tembe, Jill Muehling, John V Pearson, Dietrich A Stephan, Stanley F Nelson, and David W Craig. Resolving individuals contributing trace amounts of dna to highly complex mixtures using high-density snp genotyping microarrays. *PLoS genetics*, 4(8):e1000167, 2008.
9. Yuzhou Jiang, Tianxi Ji, Pan Li, and Erman Ayday. Reproducibility-oriented and privacy-preserving genomic dataset sharing. *arXiv preprint arXiv:2209.06327*, 2022.
10. Arthur Korte and Ashley Farlow. The advantages and limitations of trait analysis with gwas: a review. *Plant methods*, 9:1–9, 2013.
11. Xihong Lin. Reproducibility and replicability in large scale genetic studies. *Committee on reproducibility and replicability in science, National Academies of Sciences. Engineering, and Medicine*, 2018.
12. Muhammad Naveed, Erman Ayday, Ellen W Clayton, Jacques Fellay, Carl A Gunter, Jean-Pierre Hubaux, Bradley A Malin, and Xiaofeng Wang. Privacy in the genomic era. *ACM Computing Surveys (CSUR)*, 48(1):1–44, 2015.
13. Nih genomic data sharing policy. <https://sharing.nih.gov/genomic-data-sharing-policy>, 2024. [Online; accessed October-22-2024].
14. National Academies of Sciences, Policy, Global Affairs, Board on Research Data, Information, Division on Engineering, Physical Sciences, Committee on Applied, Theoretical Statistics, Board

- on Mathematical Sciences, et al. *Reproducibility and replicability in science*. National Academies Press, 2019.
15. The opensnp project. <https://opensnp.org/>, 2024. [Online; accessed October-22-2024].
 16. Sriram Sankararaman, Guillaume Obozinski, Michael I Jordan, and Eran Halperin. Genomic privacy and limits of individual detection in a pool. *Nature genetics*, 41(9):965–967, 2009.

Author Index

A

Acar, Gunes	224
Akil, Mahdi	137
Ayday, Erman	241
Azizzadeh, Shiva	224

B

Balioglu, Berkay Kemal	52
Bras-Amorós, Maria	103
Błażkiewicz, Przemysław	166

C

Cai, Bozhan	178
Castellà-Roca, Jordi	207
Chamikara, M.A.P.	18
Costa, Gabriele	69

D

Daniels, Samuel Opoku	190
Dinh, Anh	120
Doss, Robin	120

E

Evans, Robert-William	154
-----------------------	-----

F

Ferré-Queralt, Joan	207
---------------------	-----

G

Grobler, Marthie	120
Gursoy, M. Emre	52

H

Henseler, Norbert	120
-------------------	-----

J

Jiang, Yuzhou	241
---------------	-----

K

Kammüller, Florian	154
Khodaie, Alireza	52
Kikuchi, Hiroaki	35
Kubiak, Przemysław	166
Kutyłowski, Mirosław	166

L	
Lauks-Dutka, Anna	166
Lessage, Xavier	69
M	
Massonet, Philippe	69
Mbodj, Madické Diadji	86
Mitseva, Asya	178
Mouratidis, Haralambos	190
Murakami, Takao	35
O	
Orduz Álvarez, Nicolás	207
P	
Pallas, Frank	137
Panchenko, Andriy	178
Pham, Long	1
R	
Rakotoarivelo, Thierry	18
Ranbaduge, Thilina	18
S	
Sajjad, Hamza	69
Smith, David	18
T	
Torra, Vicenç	103
Truex, Stacey	1
Tsuji, Takao	207
V	
Viejo, Alexandre	207
W	
Wang, Qing	18
Wu, Tingmin	120
X	
Xu, Xincheng	18
Xu, Yuqiao	241