

# **DPM 2025 Pre-proceedings**

## 20th International Workshop on Data Privacy Management

September 25, 2025  
Toulouse, France

# Contents

<b>Contents</b>	<b>1</b>
<b>Session 1: Privacy &amp; AI</b>	<b>4</b>
How Worrying Are Privacy Attacks Against Machine Learning? . . . . .	5
Lost in the Averages: Reassessing Record-Specific Privacy Risk Evaluation . . . . .	16
Membership Inference Attacks Beyond Overfitting . . . . .	32
"Why is the sky blue?" - On the feasibility of privacy-friendly conversational LLM smart toys	49
Win-k: Improved Membership Inference Attacks on Small Language Models . . . . .	66
<b>Session 2: Applied Cryptography &amp; Statistics</b>	<b>78</b>
Advanced Electronic Signatures and GDPR: Reconciling the Concepts . . . . .	79
Invisible Encryption . . . . .	96
Eliminating Exponential Key Growth in PRG-Based Distributed Point Functions . . . . .	112
A Pseudo-Inverse Matrix-Based LDP for High-Dimensional Data . . . . .	121
Using Prior Knowledge to Improve GANs for Tabular Data Without Compromising Privacy .	133
<b>Session 3: Applications</b>	<b>150</b>
Lessons from a Robotaxi: Challenges in Selecting Privacy-Enhancing Technologies . . . . .	151
Performance Analysis of Lightweight Transformer Models for Healthcare Application Privacy	
Threat Detection . . . . .	168
The Bitter Pill: Tracking and Remarketing on EU Pharmacy Websites . . . . .	185
PADOME: Adaptive Privacy Assistant for the Internet of Things . . . . .	202

# 19th International Workshop on Data Privacy Management

## PC Chairs

Joaquin Garcia-Alfaro (Institut Polytechnique de Paris)  
Guillermo Navarro-Arribas (Universitat Autònoma de Barcelona)

## Program Committee

Abderrahim Ait Wakrime (UM5R)  
Ken Barker (University of Calgary)  
Elisa Bertino (Purdue University)  
Alessandro Brighente (University of Padova)  
Jordi Casas-Roma (Universitat Autònoma de Barcelona)  
Jordi Castellà-Roca (Universitat Rovira i Virgili)  
Depeng Chen (Anhui University)  
Mathieu Cunche (University of Lyon / Inria)  
Frédéric Cuppens (Polytechnique Montreal)  
Sabrina De Capitani di Vimercati (Università degli Studi di Milano, Italy)  
Jose Maria de Fuentes (Universidad Carlos III de Madrid)  
Josep Domingo-Ferrer (Universitat Rovira i Virgili)  
Lorena González Manzano (Universidad Carlos III de Madrid)  
M. Emre Gursoy (Department of Computer Engineering, Koç University, Istanbul, Turkey)  
Guy-Vincent Jourdan (University of Ottawa)  
Florian Kammueler (Middlesex University London and TU Berlin)  
Bruce Kapron (University of Victoria)  
Sokratis Katsikas (Norwegian University of Science and Technology)  
Christophe Kiennert (Télécom SudParis)  
Hiroaki Kikuchi (Meiji University)  
Evangelos Kranakis (Carleton University, Computer Science)  
Romain Laborde (Université de Toulouse)  
Patrick Lacharme (Ensicaen)  
Giovanni Livraga (University of Milan)  
Brad Malin (Vanderbilt University)  
Lukas Malina (Brno University of Technology)  
Zoltan Mann (University of Halle-Wittenberg)  
David Megías (UOC)  
Gerardo Pelosi (Politecnico di Milano)

Cristina Pérez-Solà (Universitat Autònoma de Barcelona)  
Ruben Rios (University of Málaga)  
Julián Salas (Internet Interdisciplinary Institute, Universitat Oberta de Catalunya)  
Pierangela Samarati (Università degli Studi di Milano, Italy)  
Vicenc Torra (Umeå University)  
Alexandre Viejo (Universitat Rovira i Virgili)  
Isabel Wagner (University of Basel)  
Jens Weber (University of Victoria)  
Nicola Zannone (Eindhoven University of Technology)


### **Additional Reviewers**

Sergio Martinez  
Pablo Sanchez-Serrano  
Carles Anglès-Tafalla  
Cristofol Dauden-Esmel  
Rami Haffar



## **Session 1: Privacy & AI**

# How Worrying Are Privacy Attacks Against Machine Learning?

Josep Domingo-Ferrer<sup>1,2</sup> 

<sup>1</sup> Universitat Rovira i Virgili,  
Department of Computer Engineering and Mathematics,  
CYBERCAT-Center for Cybersecurity Research of Catalonia,  
Av. Països Catalans 26, 43007 Tarragona, Catalonia  
`josep.domingo@urv.cat`

<sup>2</sup> LAAS-CNRS, Université de Toulouse  
7 Av. du Colonel Roche, 31400 Toulouse, France

**Abstract.** In several jurisdictions, the regulatory framework on the release and sharing of personal data is being extended to machine learning (ML). The implicit assumption is that disclosing a trained ML model entails a privacy risk for any personal data used in training comparable to directly releasing those data. However, given a trained model, it is necessary to mount a *privacy attack* to make inferences on the training data. In this concept paper, we examine the main families of privacy attacks against predictive and generative ML, including membership inference attacks (MIAs), property inference attacks, and reconstruction attacks. Our discussion shows that most of these attacks seem less effective in the real world than what a *prima facie* interpretation of the related literature could suggest.

**Keywords:** Machine learning; privacy; discriminative models; generative models; membership inference attacks; property inference attacks; reconstruction attacks.

## 1 Introduction

The main regulations in the EU that affect the development of AI are the General Data Protection Regulation (GDPR) and the EU Artificial Intelligence Act. Both were conceived before the boom of generative AI in 2022. Furthermore, the EU has announced the implementation of a Code of Practice for general purpose AI models [28]. Outside Europe, in 2023 President Biden had signed Executive Order 14110, which committed the USA to a strong regulation of the development and use of AI along similar lines as the European regulation. However, in 2025 President Trump has signed Executive Order 14179, which basically revokes Bidens order and removes all AI regulations, allegedly to “remove barriers to American leadership in artificial intelligence”.

Given the above situation and the fact that Chinese regulations on AI are more focused on protecting the government than the citizens from AI, the EU remains the world’s only major economic bloc committed to trustworthy AI. At the same time, the EU lags behind the USA and China from the AI technology point of view.

For the European AI industry to be able to catch up with its competitors in spite of a more strict regulatory framework, it is extremely important to make sure that regulations are not more strict than required to preserve the values of trustworthy AI, and in particular privacy. Unfortunately, this does not seem to be the case today. The EU regulations assume that *any* disclosure might cause a breach of privacy. In particular, there is an implicit assumption that the disclosure of a

trained machine learning (ML) model entails a privacy risk for any personal data used in training comparable to the direct release of those data.

This overcautious approach is probably due to the rushed inclusion of generative AI in the legal texts, and it may lead to adopting countermeasures that increase the training overhead and decrease the accuracy of models. For example, differential privacy [12] is a commonly proposed countermeasure that can cause two-digit drops in model accuracy if applied with meaningful privacy parameters. This seriously compromises the performance and competitiveness of the models and might be *unnecessary* if risks can be demonstrated to be overestimated.

## Contribution and plan of this article

There is a fundamental privacy difference between releasing an ML model trained on personal data and directly releasing those training data. If only the trained ML model is disclosed, it is necessary to mount a *privacy attack* to make any inferences on the training data. In this paper, we discuss how effective the privacy attacks proposed in the literature against predictive and generative ML are in *real-world* conditions. Specifically, we cover membership inference attacks (MIAs), property inference attacks, and reconstruction attacks.

Our assessment concentrates on the disclosure potential of those attacks at the conceptual level, rather than on the analysis of the internals of the various attack techniques. We aim to uncover fundamental limitations of privacy attacks.

Section 2 gives a background on privacy disclosure. Section 3 is devoted to membership inference attacks. Section 4 discusses property inference attacks. Section 5 deals with reconstruction attacks. Conclusions are drawn in Section 6.

## 2 Background on privacy disclosure

For many years, the literature on database privacy [17] has used the notion of disclosure risk, in order to measure to what extent the release of data sets and statistical output puts sensitive information at risk of being disclosed. This notion remains relevant in the machine learning domain.

Two types of disclosure have usually been considered [17]:

- *Identity disclosure* means that the attacker is able to link some unidentified piece of data released with the subject (individual) to whom it corresponds. This linkage is also called *re-identification*.
- *Attribute disclosure* means that the attacker can determine the value of a confidential attribute (*e.g.*, income, diagnosis, etc.) for a target subject with great precision *after* seeing the released data.

In tabular data, reidentification occurs trivially if the released data contain *personal identifiers* (such as passport numbers). That is why identifiers should never be released. However, re-identification is also possible by *quasi-identifiers* (for example, gender, job, zipcode, age) that do not uniquely identify the subject, but whose combination may because they may be present in public identified databases such as electoral rolls. Finally, *confidential attributes* (income, diagnosis, etc.) reveal sensitive information about subjects when they can be unequivocally linked to them.

Identification and attribute disclosure can occur independently. A record can be reidentified but, if it contains no confidential attribute, no attribute disclosure occurs. Similarly, if the attacker can only determine a set of  $k > 1$  records that might correspond to the target subject, but there is a

confidential attribute whose values over those  $k$  records are very similar, then attribute disclosure has occurred without reidentification.

*Membership disclosure* has been proposed as a third type of disclosure in machine learning [26]. Its purpose is to determine whether a given data point was included in the data set used to train a certain ML model. Thus, in a membership inference attack (MIA), the attacker does not try to discover to whom the point corresponds (which would be reidentification) or to find the value of any confidential attribute about the subject to whom the point corresponds (which would be attribute disclosure).

Thus, it can be argued that membership disclosure is weaker than identity or attribute disclosure. However, *if all subjects included in the training data set are known to share a sensitive condition, attribute disclosure can result from membership disclosure*. For example, if all subjects whose data are used for training suffer from a certain disease, then discovering membership for a target subject leads to attribute disclosure: the target suffers from that disease. Note that this is true even if there was no explicit attribute ‘Disease’ in the training data set.

### 3 Membership inference attacks

MIAs are the most common attack employed to assess the privacy of training data in machine learning. Rather than analyzing the operation of specific MIAs proposed in the literature, in this section we will focus on the disclosure potential of a generic MIA depending on the data used to train the model under attack.

Let us introduce a running example. Assume that an attacker Alice wants to perform an MIA on an ML model to determine whether the attacker’s neighbor Neil was a member of the data used to train the model.

Two properties of the training data are simultaneously required for the MIA to allow unequivocal inferences:

- *Exhaustivity*. Unless the training data were an exhaustive sample of a population (which is very rare), membership inferences cannot be unequivocal. In other words, the membership revealed by an MIA to a non-exhaustive training set could be plausibly denied. In the running example, if Alice finds that one or several records containing the same quasi-identifier values known to her about Neil were members of the training data, she cannot be absolutely sure that Neil was a member. The reason is that perhaps Neil was not included in the training data, and the putative members she found are just people sharing Neil’s quasi-identifier values. Hence, Neil could plausibly deny being a member. On the other hand, if the training data set is exhaustive, membership is trivial and no MIA is really needed: every existing record (and Neil’s in particular) is a member.
- *Non-diversity of unknown attributes*. Since inferring membership to an exhaustive sample is not a real discovery, let us examine whether at least it can bring attribute disclosure. If there are several member records matching the attribute values known to the attacker, and the unknown attributes among those records differ significantly, then no attribute disclosure occurs. In the running example, if Alice finds that two or more records containing the same quasi-identifiers known to her about Neil were members of the (exhaustive) training data, but the confidential attribute Income unknown to Alice take clearly different values on those records, then Alice cannot unequivocally learn Neil’s income.

In summary, the training data must be exhaustive for Alice to be sure that at least one of the putative members she has found who share Neil’s quasi-identifiers is really Neil. On the other hand, since membership inference to an exhaustive sample is of little value, if Alice turns to attribute inference, she can unequivocally infer a confidential attribute value for Neil only if all putative members sharing Neil’s quasi-identifiers share the same (or similar) values for that confidential attribute. In looking at the literature on MIAs, most attacks are demonstrated using training data sets that are not exhaustive and that may contain diverse values for unknown attributes.

The two conditions have long been studied in the statistical disclosure control (SDC) literature [17]:

- The protective effect of non-exhaustive samples is the principle of a well-known SDC method called sampling, in which a sample is released instead of the entire surveyed population. To evaluate the protection provided by sampling, it is relevant to compute the probability that a record is unique in the population ( $PU$ ) given that it is unique in the sample ( $SU$ ), that is,  $\Pr(PU|SU)$ . In [27] it was shown that this probability decreases with the sampling fraction, that is, the smaller the sample, the more plausible membership deniability.
- The protective effect of diversity against confidential attribute disclosure is the principle behind privacy models such as  $l$ -diversity [21] and  $t$ -closeness [19]: both seek to prevent attribute disclosure by making sure there is enough diversity of confidential attribute values within each set of records sharing quasi-identifier values.

In fact, it is relatively easy for a model trainer to benefit from the above two protections against MIAs. It is easier to get non-exhaustive than exhaustive training data, and the latter can always be made non-exhaustive by sampling. On the other hand, confidential attributes are naturally diverse and, if there is not enough diversity, it can be enforced by  $l$ -diversity or  $t$ -closeness.

Beyond studying the generic limitations of MIAs due to the data used to train the attacked ML models, one can examine the specifics of the model under attack and the attack method. That is, what else is needed for an MIA to succeed in the case when the training data happen to satisfy exhaustivity and non-diversity.

In [18], the effectiveness of MIAs on discriminative machine learning (ML) models is assessed by checking four requirements: i) the model under attack should not be overfitted (overfitted models are an easy MIA target, but they do not generalize well in their main tasks); ii) the model under attack must have a competitive test accuracy (attacking an uncompetitive model is not very interesting); iii) the attack must yield reliable membership inference; iv) and the attack must have a reasonable computational cost. Among the many MIA attacks reviewed by these authors, none can satisfy these four requirements simultaneously.

In fact, focusing only on overfitting, [9] had previously observed that MIAs on well-generalizable models suffer from practical limitations that reduce their practicality. Overall, it would seem that the privacy risks of machine learning may have been overstated in the literature as far as membership inference attacks are concerned.

## 4 Property inference attacks

A property inference attack seeks to infer a sensitive *global* property of the data set used to train an ML model, that is, a property  $P$  of the data set that the model producer did not intend to share. This class of attacks was first presented for classifiers in [1]. They have also been formulated for deep neural networks in [15].

In [1], a meta-classifier was trained to classify the target classifier depending on whether it has a certain property  $P$  or not. To do this, the attacker trains several *shadow classifiers* on the same task as the target classifier. Each classifier is trained on a data set similar to that of the target classifier, but constructed explicitly to have the property  $P$  or not. Subsequently, the meta-classifier is trained on the sets of parameters of the shadow classifiers.

In [15], it is argued that the above meta-classifier training strategy does not work well for deep neural networks, due to their complexity and thousands of parameters. The authors explore different feature representations to reduce the complexity of the meta-classification task. However, the high-level structure of the attack is the same as in [1]. In [31], a property inference attack against generative adversarial networks (GANs) is presented. Instead of training shadow classifiers like in the previous papers, here shadow GANs are trained.

From the point of view of privacy, property inference attacks do not entail a significant risk, because *they aim to infer a general property of the training data set rather than a property specific to a particular target subject*. That is, *property inference attacks are not attribute inference attacks* trying to infer the value of a confidential attribute for a target subject. By way of illustration, an example of  $P$  mentioned in [1] is whether “Google traffic was used in the training data”, an example mentioned in [15] is whether “the classifier was trained on images with noise”, and an example mentioned in [31] is whether “a GAN is mainly trained with images of white males”.

Even if disclosing such properties was not intended by the model producer and may cause some embarrassment to them, the general nature of those properties can hardly disclose private information on any of the specific individuals whose data may have been used for training. The most obvious strategy to mount an attribute inference attack in machine learning is through a battery of MIAs each of which hypothesizes a candidate value for the target subject’s confidential attribute (*e.g.*, was the target subject’s record with “Disease=AIDS” a member of the training data set? was the target subject’s record with “Disease=Cancer” a member of the training data set?, and so on).

A scenario where property inference attacks may be more privacy-disclosive is federated learning, in case they are used to infer properties of the training data set used by a certain client and those training data refer to just one or a few subjects. Imagine the client is a smartphone and the client’s training data are health measurements on the smartphone owner at different times; in this specific case, inferring a property of the training data set can yield a property/attribute of the smartphone owner.

## 5 Reconstruction attacks

### 5.1 Reconstruction attacks previous to ML

Dinur and Nissim (DN from now on) developed a formal theory of database reconstruction from a set of query responses in 2003 [8]. The authors assume that a database is an  $n$ -bit string, that is, it contains records each of which takes values 0 or 1. They further assume all queries to be of the form “How many records in this subset are 0’s?” or “How many records in this subset are 1’s?”. In their setting, the response to every query is computed as the true answer to the query plus an error  $E$  bounded in an interval  $[-B, B]$  for some  $B > 0$ . Thus, the assumption is that query answers are protected by output perturbation with strictly bounded noise.

According to DN, a database reconstruction is a record-by-record reconstruction of the original values such that the distance between the reconstructed values and the original values is within

specific accuracy bounds. DN considered two types of attackers, one that can ask an exponential number of queries and one that can only ask a polynomial number of queries, and gave results for the reconstructions achievable by those attackers as a function of  $B$  and the number of queries allowed. Although such a theoretical framework for database reconstruction provides very relevant insight, it does not mean that every database can be uniquely reconstructed. In fact, for a given set of statistical outputs, there may be several (or even a large number) of database instantiations compatible with those outputs [22,24].

## 5.2 Reconstruction attacks and overfitting in ML

The problem of reconstructing the data set used to train an ML model bears some similarities to the database reconstruction problem just described. During machine learning, sometimes the model memorizes parts of its training data [13]. This in turn enables attackers to extract points from the training data set when given access to the trained model. Successful reconstruction attacks have been reported for face recognition models [14,30] and neural language models [6,7]. Although there is no formal framework in the DN style for reconstruction in ML, bounds on the risk of reconstruction have been proven [16].

In fact, (partial) reconstruction of training data is greatly facilitated if the model is overfitted because, in that case, it memorizes training data. Beyond being problematic for privacy, overfitting is also a great problem for utility, since overfitted models usually perform poorly regarding validation (the process of testing how well a trained model labels new, unseen data).

Regarding potential defenses against overfitting and, hence, reconstruction, [6] mention that

“such memorization [of training data] is *not* due to overtraining: it occurs early during training, and persists across different types of models and training strategies [...] Furthermore, we show that simple, intuitive regularization approaches such as early-stopping and dropout are insufficient to prevent unintended memorization. Only by using differentially-private training techniques, we are able to eliminate the issue completely, albeit at some loss of utility.”

Overtraining means training a model for too many iterations. It may result in overfitting, which occurs when the model exactly learns the training data set but is unable to correctly label new, unseen data. However, overfitting may also occur in the early stages of training, that is, without overtraining, such as when a very large model is trained on a small data set.

In [3], it was concluded that standard anti-overfitting techniques such as regularization and dropout could outperform DP and achieve a better utility/privacy/efficiency trade-off in ML training. The explanation of this seeming contradiction with [6] lies in the details:

- [3] tried several combinations of regularization/dropout and took the one with the best trade-off between utility, measured as test accuracy, and privacy, measured as the attacker’s (little) advantage in the standard MIA implementation in TensorFlow Privacy.
- In contrast, [6] tried several anti-overfitting techniques (regularization, dropout, weight quantization, etc.) but without attempting to find the best-performing parameterizations. Also, they measured utility as (little) validation loss and privacy as preventing the recovery of randomly chosen “canary” sequences inserted into the models’ training data.

Regarding utility, note that test accuracy and validation loss are two independent metrics. Whereas the former counts the number of mistakes/misclassifications, the latter is the distance

between the true labels and the labels predicted by the model. Low test accuracy means many errors, whereas large validation loss means large errors.

Regarding privacy, the two above papers and a good deal of the related literature use MIA-based metrics. There are two important factors that influence the success of MIAs: (i) whether the target points whose membership is to be inferred are outliers or not and (ii) how good the MIA techniques employed are. Now, the random “canary” target sequences inserted by [6] in the training data are likely to be outliers due to their randomness, and hence their membership may be easy to discover, which gives a pessimistic privacy evaluation. The TensorFlow Privacy MIA implementation used by [3] does not rely on the inserting of random target points into the training data: it just uses the predictions of the trained model on the target points to deduce their membership [4].

### 5.3 On the effectiveness of reconstruction in ML

Using MIAs to assess the effectiveness of reconstruction attacks may seem reasonable if the training data are tabular. Let  $\mathbf{D}$  be a training data set with attributes  $A_1, A_2, \dots, A_d$ . Note that in the computer representation of any attribute  $A_i$ , the number  $|A_i|$  of potential values can be considered finite, even for numerical attributes, due to limited length and precision. Still,  $|A_i|$  can be quite large, especially for numerical attributes. We can give the following information-theoretic argument to illustrate the complexity of exhaustively trying all possible values. Assume that the information content of an item  $X$  (record in the case of tabular data, but also unstructured text, image, etc., for non-structured data) one wishes to reconstruct is  $H(X)$  bits, where  $H$  is Shannon’s entropy. Then discovering  $X$  by exhaustive search is equivalent to discovering a random cryptographic key of  $H(X)$  bits. If  $H(X)$  is, say, 64 or more bits, this is known to be computationally infeasible.

This gives two scenarios:

1. *Total reconstruction.* Assume that the attacker has unlimited resources or, better, that the number of potential values  $|A_i|$  of every attribute  $A_i$  is relatively small. In this case, the attacker could mount an MIA for each possible combination of attribute values, to check whether that combination was part of  $\mathbf{D}$ . After  $\prod_{i=1}^d |A_i|$  MIAs have been performed and if they are effective, the attacker has reconstructed the entire training data set  $\mathbf{D}$ .
2. *Partial reconstruction.* If the attacker’s resources are insufficient to pursue total reconstruction, then they can select a subset of possible combinations of attribute values and mount MIAs only for those combinations. This can be viewed as a *guessing exercise* that may lead to a partial reconstruction of  $\mathbf{D}$  (if the guesses, that is, the candidate combinations of attribute values, are classified as members and are really members of  $\mathbf{D}$ ). The attacker would favor those combinations deemed to be the most likely from the semantics of attributes, *e.g.* if there is an attribute *Age* and an attribute *Job*, the only plausible combination of *Age*=10 is with *Job*=‘student’. Note that betting on the most common combinations gives less interesting reconstruction results for the attacker: outlier combinations are more privacy-sensitive and thus interesting to the attacker than very common combinations.

It must be taken into account that state-of-the-art MIAs offering the best membership detection, such as LiRA [5], require training several shadow models to estimate the distribution  $\Delta_{in}$  of models trained on data sets containing the target point and the distribution  $\Delta_{out}$  of models trained on data sets *not* containing the target point. Thus, each MIA incurs a substantial computation cost.

Furthermore, especially in generative ML, training data are often non-tabular. For example, they are unstructured text or multimedia. Clearly, for non-tabular training data such as images



or unstructured text, mounting an MIA to test whether each potential image or each potential unstructured text was part of the training data set  $\mathbf{D}$  seems quite unreasonable. In the case of generative AI, one can resort to prompting for certain personal data or copyrighted content rather than mounting MIAs, in order to find out whether the model saw those items at training time. But in fact, this prompting amounts to a guessing exercise like those described above under partial reconstruction. In fact, the empirical study [11] shows that MIAs on pre-trained LLMs are barely better than random guessing, even though fine-tuned LLMs are far more vulnerable to MIAs. That is, MIAs are more effective at inferring membership on the data used for fine-tuning than on the data used for pre-training. Regarding cost, although guess prompting is almost free on the user’s side, the computational cost is high in terms of LLM inference on the LLM manager’s side.

In [20] a systematic evaluation of data reconstruction attacks and defenses is presented, where the reconstruction attacks considered are no longer MIAs, but *gradient inversion attacks*. Gradient inversion attacks [29] attempt to recover training points from gradients. They are mostly designed for federated learning (FL), because they require knowledge of the gradients computed during training. In fact, in FL, the server receives the gradients from the clients and can mount a gradient inversion attack and try to reconstruct the local training data for one or more clients [25]. If all clients receive all gradients, then clients can also behave maliciously and mount a gradient inversion attack to reconstruct the local data of a certain target client. The study [23] reviewed gradient inversion attacks against FL, as well as potential defenses based on mixed precision and quantization, gradient pruning, and differential privacy. They concluded that some of these defenses are effective and involve only slight accuracy drops. In centralized learning, where the attacker only sees the trained model, gradient inversion attacks are not applicable.

If reconstruction based on MIAs is problematic for the reasons above, reconstruction without MIAs suffers from a major weakness: *there is no numerical decision criterion in a realistic case in which the attacker has no access to the actual training data*. In other words, whereas in an MIA there is some kind of threshold that allows deciding whether a target point is a member or a non-member (although this decision may be in error), in a reconstruction attack there is no objective criterion to decide whether the putative reconstructed data belong to the training data set. For example, the fact that a gradient inversion attack produces a meaningful image does not necessarily mean that this image was part of the training data. Also, what “meaningful” means is debatable. One could certainly use an MIA to decide whether the putative reconstructed data were really in the training data, but this has the drawbacks of MIAs enumerated in Section 3.

Admittedly, there are situations in which it may be easier to make a decision on putative reconstructed data. This is the case for reconstruction attacks on machine unlearning. In unlearning, a trained model is updated to cause it to “forget” one or more data points, *e.g.*, to implement the right to be forgotten enshrined in the GDPR, or because those data points are subject to copyright. In [2], a reconstruction attack is described for the case in which the trained model is a simple one. The attack exploits the model updates to estimate the unlearned data point. However, even if the attack is quite successful according to the experiments reported in [2], success is determined by comparing against the ground truth of the unlearned data point, which would not be available to an attacker in a real world situation. Possible defenses are discussed in [10].

## 6 Conclusions

Our analysis casts doubts on the effectiveness of privacy attacks against ML in real-world conditions:

- MIAs suffer from limitations due to the data the target models have been trained on (non-exhaustivity, diversity of confidential attribute values). In addition, they may also suffer limitations that arise from the nature of the attacked models and the attack methods.
- Property inference attacks aim to infer a general property of the training data set, rather than a property specific to a particular data subject. For that reason, they do not achieve attribute disclosure for any particular subject and hence do not pose substantial privacy risks to subjects, except in specific federated learning scenarios where all of a client’s training data refer to one or a few subjects. These attacks are more relevant to audit the potential biases or insufficiencies of the training data used by the model producer.
- Reconstruction attacks based on MIAs have a very significant cost, as they involve mounting an MIA for each data point whose membership in the training data is to be decided. Thus, they are only practical for tabular training data where attributes have a limited range of potential values, and even in that case they are more suited for partial than total reconstruction. Besides, MIA-based reconstruction is also subject to the shortcomings identified for MIAs themselves.
- Reconstruction attacks based on gradient inversion are those that are used when training data are multimedia or unstructured text, as is the usual case in generative ML. However, such attacks are applicable only when the attacker has access to the gradients computed by the victim during the learning process. In practice, this restricts the applicability of these attacks to federated or otherwise decentralized learning. Furthermore, deciding whether a putative reconstructed data point was really a member of the training data is difficult if the attacker does not have access to the original training data (which is the usual case in the real world). Certainly, MIAs can be used to make this membership decision, but this inherits the shortcomings of MIAs described above.

All in all, the current real-world privacy risks incurred by machine learning seem less serious than what is usually assumed in the literature. Therefore, privacy defenses that entail severe utility loss, such as differential privacy, may be often unnecessary. The good side of all this is that trustworthy machine learning may be easier to implement than assumed so far, at least with respect to privacy. This is good news for jurisdictions like the European Union that struggle to reconcile strong AI regulations with the competitiveness of their AI industry.

## Acknowledgments

This work was partly funded by the Centre International de Mathématiques et d’Informatique de Toulouse (CIMI), the Government of Catalonia (ICREA Acadèmia Prize to J. Domingo-Ferrer), MCIN/AEI/ 10.13039/501100011033 and “ERDF A way of making Europe” under grant PID2021-123637NB-I00 “CURLING”, and INCIBE and European Union NextGenerationEU/PRTR (project “HERMES” and INCIBE-URV Cybersecurity Chair).

## References

1. Giuseppe Ateniese, Luigi V Mancini, Angelo Spognardi, Antonio Villani, Domenico Vitali, and Giovanni Felici. Hacking smart machines with smarter ones: How to extract meaningful data from machine learning classifiers. *International Journal of Security and Networks*, 10(3):137–150, 2015.
2. Martin Bertran, Shuai Tang, Michael Kearns, Jamie H Morgenstern, Aaron Roth, and Steven Z Wu. Reconstruction attacks on machine unlearning: Simple models are vulnerable. *Advances in Neural Information Processing Systems*, 37:104995–105016, 2024.

3. Alberto Blanco-Justicia, David Sánchez, Josep Domingo-Ferrer, and Krishnamurty Muralidhar. A critical review on the use (and misuse) of differential privacy in machine learning. *ACM Computing Surveys*, 55(8):1–16, 2022.
4. Franziska Boenisch. Attacks against machine learning privacy (part 2): membership inference attacks with TensorFlow Privacy. <https://franziska-boenisch.de/posts/2021/01/membership-inference/>, 2021. (Accessed on 14/05/2025).
5. Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramer. Membership inference attacks from first principles. In *2022 IEEE symposium on security and privacy (SP)*, pages 1897–1914. IEEE, 2022.
6. Nicholas Carlini, Chang Liu, Úlfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: Evaluating and testing unintended memorization in neural networks. In *28th USENIX security symposium (USENIX security 19)*, pages 267–284, 2019.
7. Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Úlfar Erlingsson, et al. Extracting training data from large language models. In *30th USENIX security symposium (USENIX Security 21)*, pages 2633–2650, 2021.
8. Irit Dinur and Kobbi Nissim. Revealing information while preserving privacy. In *Proceedings of the twenty-second ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 202–210, 2003.
9. Antreas Dionsysiou and Elias Athanasopoulos. Sok: Membership inference is harder than previously thought. *Proceedings on Privacy Enhancing Technologies*, 2023.
10. Josep Domingo-Ferrer, Najeeb Jebreel, and David Sánchez. Defenses against membership inference attacks on unlearned data. In *Modeling Decisions in Artificial Intelligence (MDAI 2025)*. Springer (to appear), 2025.
11. Michael Duan, Anshuman Suri, Niloofar Miresghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841*, 2024.
12. Cynthia Dwork, Frank McSherry, Kobbi Nissim, and Adam Smith. Calibrating noise to sensitivity in private data analysis. In *Theory of Cryptography: Third Theory of Cryptography Conference, TCC 2006, New York, NY, USA, March 4-7, 2006. Proceedings 3*, pages 265–284. Springer, 2006.
13. Vitaly Feldman. Does learning require memorization? a short tale about a long tail. In *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*, pages 954–959, 2020.
14. Matt Fredrikson, Somesh Jha, and Thomas Ristenpart. Model inversion attacks that exploit confidence information and basic countermeasures. In *Proceedings of the 22nd ACM SIGSAC conference on computer and communications security*, pages 1322–1333, 2015.
15. Karan Ganju, Qi Wang, Wei Yang, Carl A Gunter, and Nikita Borisov. Property inference attacks on fully connected neural networks using permutation invariant representations. In *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*, pages 619–633, 2018.
16. Chuan Guo, Brian Karrer, Kamalika Chaudhuri, and Laurens van der Maaten. Bounding training data reconstruction in private (deep) learning. In *International Conference on Machine Learning*, pages 8056–8071. PMLR, 2022.
17. Anco Hundepool, Josep Domingo-Ferrer, Luisa Franconi, Sarah Giessing, Eric S. Nordholt, Keith Spicer, and Peter-Paul De Wolf. *Statistical Disclosure Control*. Wiley, Chichester UK, 2012.
18. Najeeb Jebreel, David Sánchez, and Josep Domingo-Ferrer. A critical review on the effectiveness and privacy threats of membership inference attacks. *submitted manuscript*, 2025.
19. Ninghui Li, Tiancheng Li, and Suresh Venkatasubramanian.  $t$ -Closeness: Privacy beyond  $k$ -anonymity and  $l$ -diversity. In *2007 IEEE 23rd international conference on data engineering*, pages 106–115. IEEE, 2006.
20. Sheng Liu, Zihan Wang, Yuxiao Chen, and Qi Lei. Data reconstruction attacks and defenses: A systematic evaluation. *arXiv preprint arXiv:2402.09478*, 2025.
21. Ashwin Machanavajjhala, Daniel Kifer, Johannes Gehrke, and Muthuramakrishnan Venkitasubramanian.  $l$ -Diversity: Privacy beyond  $k$ -anonymity. *Acm transactions on knowledge discovery from data (tkdd)*, 1(1):3-es, 2007.

22. Krishnamurty Muralidhar and Josep Domingo-Ferrer. Database reconstruction is not so easy and is different from reidentification. *Journal of Official Statistics*, 39(3):381–398, 2023.
23. Pretom Roy Ovi and Aryya Gangopadhyay. A comprehensive study of gradient inversion attacks in federated learning and baseline defense strategies. In *2023 57th Annual Conference on Information Sciences and Systems (CISS)*, pages 1–6. IEEE, 2023.
24. David Sánchez, Josep Domingo-Ferrer, and Krishnamurty Muralidhar. Confidence-ranked reconstruction of census records from aggregate statistics fails to capture privacy risks and reidentifiability. *Proceedings of the National Academy of Sciences*, 120(18):e2303890120, 2023.
25. Yichuan Shi, Olivera Kotevska, Viktor Reshniak, Abhishek Singh, and Ramesh Raskar. Dealing doubt: Unveiling threat models in gradient inversion attacks under federated learning, a survey and taxonomy. *arXiv preprint arXiv:2405.10376*, 2024.
26. Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pages 3–18. IEEE, 2017.
27. CJ Skinner. Disclosure avoidance for census microdata in great britain. In *Proceedings of the 1990 Annual Research Conference*, pages 131–143. US Bureau of the Census, 1990.
28. European Union. General-purpose ai code of practice. <https://digital-strategy.ec.europa.eu/en/policies/ai-code-practice>, 2025. (Accessed on 26/05/2025).
29. Rui Zhang, Song Guo, Junxiao Wang, Xin Xie, and Dacheng Tao. A survey on gradient inversion: Attacks, defenses and future directions. *arXiv preprint arXiv:2206.07284*, 2022.
30. Yuheng Zhang, Ruoxi Jia, Hengzhi Pei, Wenxiao Wang, Bo Li, and Dawn Song. The secret revealer: Generative model-inversion attacks against deep neural networks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 253–261, 2020.
31. Junhao Zhou, Yufei Chen, Chao Shen, and Yang Zhang. Property inference attacks against gans. *arXiv preprint arXiv:2111.07608*, 2021.

# Lost in the Averages: Reassessing Record-Specific Privacy Risk Evaluation

Nataša Krčo<sup>1</sup>, Florent Guépin<sup>1</sup>, Matthieu Meeus<sup>1</sup>, Bogdan Kulynych<sup>2</sup>, and  
Yves-Alexandre de Montjoye<sup>1</sup>

<sup>1</sup> Department of Computing and Data Science Institute, Imperial College London,  
London, United Kingdom

{n.krco23,florent.guepin20, m.meeus22, deMontjoye}@imperial.ac.uk

<sup>2</sup> Lausanne University Hospital (CHUV), Lausanne, Switzerland  
bogdan.kulynych@chuv.ch

**Abstract.** Synthetic data generators and machine learning models can memorize their training data, posing privacy concerns. Membership inference attacks (MIAs) are a standard method of estimating their privacy risk. The risk of individual records is typically computed by evaluating MIAs in a record-specific privacy game. We analyze the privacy game commonly used for attackers under realistic assumptions (the *traditional* game)—particularly for synthetic tabular data—and show that it averages a record’s privacy risk across datasets. We show this implicitly assumes the dataset a record is part of has no impact on the record’s risk, providing a misleading risk estimate when a specific model or synthetic dataset is released. Instead, we propose a novel use of the leave-one-out privacy game, so far used exclusively to audit differential privacy guarantees, and call this the *model-seeded* game. We formalize it and show that it provides an accurate estimate of the privacy risk for a record in its specific dataset. We instantiate and evaluate the state-of-the-art MIA for synthetic data generators in both privacy games, and show across multiple datasets and models that they indeed result in different risk scores, with up to 94% of high-risk records being overlooked by the traditional game. We further show that records in smaller datasets tend to have a larger gap between risk estimates. Taken together, our results show that the model-seeded setup yields a risk estimate specific to a released synthetic dataset or model and in line with the standard notion of privacy leakage from prior work, meaningfully different from the dataset-averaged risk provided by the traditional privacy game.

**Keywords:** membership inference, synthetic data, differential privacy

## 1 Introduction

Models ranging from synthetic data generators (SDGs) to machine learning (ML) models have been shown to memorize their training data, potentially allowing

---

An extended version of this work including appendices and additional results is available at <https://arxiv.org/abs/2405.15423>.

attackers to tell whether specific records were used for training [2, 18, 20, 22, 31, 41, 46] or even reconstruct entire training examples [5, 21, 49, 53]. As models are increasingly trained on personal and sensitive data—particularly in domains such as healthcare, law, and finance [7, 10, 37]—concerns about their implications for privacy continue to grow.

Membership inference attacks (MIAs) have become the standard approach for empirically estimating the privacy risk of synthetic data and ML models [8, 28, 40, 44, 46]. MIAs aim to determine whether a target record was included in the training dataset of a given model. They can pose a direct privacy risk, and also provide an upper bound on the performance of other attacks such as attribute inference or data reconstruction [43]. MIAs can be developed under varying assumptions, ranging from black-box access to the target model and no knowledge of the training dataset, to very strong attackers leveraging white-box access to the model and knowledge of all training records but the target.

MIAs are evaluated in a controlled privacy game between an attacker and a data owner [46, 55]. We here study the record-specific privacy games used in existing literature, which estimate how well an attacker can distinguish between models trained on one specific target record and those not. Record-specific privacy games are most often used in setups where the state-of-the-art attacks leverage record-specific information, such as for synthetic data generators [22, 31, 46], and for auditing formal privacy guarantees [3]. In contrast, model-specific privacy games estimate the ability of an attacker to distinguish between records used to train one target model and those not. This type of privacy game is often used to evaluate MIAs against ML models [8, 11, 20, 42, 45, 56].

*Contributions.* We analyze the *traditional* privacy game commonly used to evaluate record-specific MIAs under realistic attacker assumptions [19, 22, 31, 46]. We show that, by using dataset sampling as a source of randomness, it averages the risk across datasets, implicitly assuming that a record’s privacy risk is independent of the dataset it belongs to.

We instead formalize and propose a novel use of the leave-one-out game, here called the *model-seeded* privacy game, to evaluate an MIA under realistic attacker assumptions. This approach is consistent with the standard notion of differential privacy [14, 15], which captures a record’s risk with respect to a specific dataset. Unlike the traditional game, we fix the target dataset and use only the model seed as a source of randomness. We show that the attack success rates computed using this privacy game converge to what we call the record’s *differential privacy distinguisher* (DPD) risk—which is consistent with the standard notion of privacy leakage in existing literature—whereas the traditional game results in a dataset-independent estimate.

We instantiate the state-of-the-art record-specific MIA for synthetic data and evaluate it in both the traditional and model-seeded privacy game across 2 datasets and 2 synthetic data generators, replicating the setup used by Meeus et al. [31]. We observe significant differences between the risk estimates given by the model-seeded and traditional privacy games. For instance, 94% of high-risk records are misidentified by the traditional privacy game for the Adult dataset

and Synthpop generator, and the root mean squared deviation (RMSD) is 0.07 between the two estimates. We obtain similar results across experimental setups.

Finally, we show the gap between the traditional and model-seeded risk estimates to be generally higher for small and medium datasets (fewer than 10,000 records), as often used in tabular synthetic data [19, 31, 46], and lower for large datasets, typically used for ML tasks [8].

Taken together, our results show that the traditional game can yield misleading estimates by averaging the risk across datasets. We propose to use instead the model-seeded privacy game which provides more accurate risk estimates, aligning with differential privacy.

## 2 Background

### 2.1 Synthetic data generation

We consider the setting of statistical learning over the space of records  $\mathbb{D} \subseteq \mathbb{R}^d$ , sampled from a probability distribution  $\mathcal{D}$ . A dataset  $D \in \mathbb{D}^n$  is i.i.d. sampled:  $D \sim \mathcal{D}^n$ . Using  $D$ , we train a model via a randomized training algorithm  $\mathcal{A} : 2^{\mathbb{D}} \rightarrow \Theta$ . The resulting synthetic data generator (SDG) defines a distribution  $\mathcal{D}_\theta$  that mimics statistical properties of  $\mathcal{D}$ . A trained SDG with parameters  $\theta = \mathcal{A}(D)$  can generate a synthetic dataset  $D_{\text{syn}} \sim \mathcal{D}_\theta^n$ , where we set  $|D_{\text{syn}}| = n$ .

SDGs include probabilistic models such as Bayesian networks [57] and deep generative models such as GANs [52]). We focus on tabular SDGs [35, 57], where record-specific evaluation is crucial as attacks for this setting are inherently record-specific (see Section 2.2).

### 2.2 MIA development

For a *target model*  $\theta = \mathcal{A}(D)$ , an MIA aims to infer whether a target record  $x$  was in  $D$  (member) or not (non-member). For a fixed target record  $x$ , we denote by  $\phi_x : \Theta \rightarrow [0, 1]$  an MIA against target record  $x$  and target model  $\theta$ . We drop the subscript  $x$  when the target record is clear from context.

*Threat model.* By *threat model*, we refer to the assumptions made about the attacker’s capabilities. We distinguish between *dataset-level* and *model-level* assumptions. Dataset-level access can range from no access to real data from  $\mathcal{D}$  [19], access to data drawn from the same distribution [22, 31, 46], or full access to  $D$  except for the knowledge of membership of  $x$ , as considered for the strong differential privacy attacker [3, 23]. Model-level access can be black-box (query access) [8, 44], or white-box (full parameter access) [12, 33, 42].

We assume a standard **realistic record-specific attacker** [19, 22, 31, 46] in the context of tabular synthetic data, with access to an *auxiliary dataset*  $D_{\text{aux}} \in 2^{\mathbb{D}}$  drawn from the same distribution as  $D$  but disjoint from it. The attacker has black-box query access to target model  $\theta$ . We assume the attacker to have full knowledge of the exact training process used to obtain  $\theta = \mathcal{A}(D)$ .

*Shadow modeling.* Shadow modeling is a technique used to develop MIAs by simulating the target model’s training process. The attacker samples *shadow datasets*  $\{D_{\text{shadow}}^{(i)} \mid i = 1 \dots, N_{\text{shadow}}\}$  from  $D_{\text{aux}}$ , of the same size as  $D$ . The attacker then explicitly constructs ‘in’ shadow datasets that include the target record  $x$  ( $x \in D_{\text{shadow}}^{(i)}$ ) and ‘out’ shadow datasets that exclude it ( $x \notin D_{\text{shadow}}^{(i)}$ ). The attacker trains *shadow models*  $\{\mathcal{A}(D_{\text{shadow}}^{(i)}) \mid i = 1, \dots, N_{\text{shadow}}\}$  using the knowledge of the training procedure of the target model. Thus, the attacker constructs a controlled set of models with known membership of the target record, which they can use to develop and refine the MIA.

*Computing a membership score.* The membership prediction of an MIA is typically in the form of thresholding a *membership score*  $s_x : \Theta \rightarrow \mathbb{R}$ . We denote the attack as  $\phi_x(\theta) = \mathbb{1}[s_x(\theta) \geq \gamma]$  for some given threshold  $\gamma \in \mathbb{R}$ .

Existing MIAs against SDGs extract features from generated data using statistical queries [22, 31], or training membership meta-classifiers per record [46]. These scores are inherently *record-specific*, driving the development and evaluation of MIAs tailored to individual records.

### 2.3 Differential privacy and its hypothesis-testing interpretation

Differential privacy (DP) is a formal privacy guarantee that limits the contribution of any single record in statistical learning. A randomized training algorithm  $\mathcal{A}(D)$  is differentially private if the inclusion or exclusion of any single record in  $D$  will not significantly modify the resulting model distribution [15]:

**Definition 1.** A randomized training algorithm  $\mathcal{A}(D)$  satisfies  $(\varepsilon, \delta)$ -DP if for any measurable subset  $E$  of the model space  $\Theta$  and any partial dataset  $\bar{D}$  and any record  $x \in \mathbb{D}$ , we have:

$$\begin{aligned} \Pr[\mathcal{A}(\bar{D}) \in E] &\leq e^\varepsilon \Pr[\mathcal{A}(\bar{D} \cup \{x\}) \in E] + \delta \\ \Pr[\mathcal{A}(\bar{D} \cup \{x\}) \in E] &\leq e^\varepsilon \Pr[\mathcal{A}(\bar{D}) \in E] + \delta \end{aligned}$$

The classical definition (Definition 1) has been shown to have an interpretation in terms of hypothesis testing [13, 25, 50], or equivalently, in terms of success rates of worst-case MIAs [26]. Consider the following MIA setting in which an adversary with access to a partial dataset  $\bar{D}$ , the target record  $x$ , and a model  $\theta$ , aims to tell whether  $\theta$  comes from  $\mathcal{A}(\bar{D})$  or  $\mathcal{A}(\bar{D} \cup \{x\})$ :

$$H_0 : \theta \sim \mathcal{A}(\bar{D}) \quad H_1 : \theta \sim \mathcal{A}(\bar{D} \cup \{x\}). \quad (1)$$

We omit the analogous case of  $H_0$  corresponding to  $\mathcal{A}(\bar{D} \cup \{x\})$  and  $H_1$  to  $\mathcal{A}(\bar{D})$ . Given a *distinguisher*  $\phi : \Theta \rightarrow [0, 1]$  which outputs 1 to guess the membership of  $x$  in the training dataset ( $H_1$ ), and 0 to guess its non-membership ( $H_0$ ), we can characterize its success by its false positive rate (FPR)  $\alpha_\phi$  and false negative rate (FNR)  $\beta_\phi$ :

$$\alpha_\phi = \mathbb{E}_{\theta \sim \mathcal{A}(\bar{D})}[\phi(\theta)], \quad \beta_\phi = 1 - \mathbb{E}_{\theta \sim \mathcal{A}(\bar{D} \cup \{x\})}[\phi(\theta)] \quad (2)$$



To analyze the privacy guarantees within this setting, we can consider the worst-case distinguisher  $\phi_\alpha^*$  which achieves the lowest FNR at a given level of FPR  $\alpha$ :

$$\phi_\alpha^* = \arg \inf_{\phi: \Theta \rightarrow [0,1]} \{\beta_\phi \mid \alpha_\phi \leq \alpha\}. \quad (3)$$

Such an optimal attack always exists and can be constructed via Neyman-Pearson's lemma [13]. An algorithm  $\mathcal{A}(\cdot)$  satisfies  $(\epsilon, \delta)$ -DP if and only if the FNR of the optimal attack is lower bounded as follows:

$$\beta_{\phi_\alpha^*} \geq \min\{0, 1 - e^\epsilon \alpha - \delta, e^{-\epsilon}(1 - \alpha - \delta)\}, \quad (4)$$

for any given level of FPR  $\alpha \in [0, 1]$ , any  $\bar{D} \in 2^{\mathbb{D}}$  and  $x \in \mathbb{D}$  [13].

We refer to the trade-off curve, i.e., the set of all attainable  $\alpha_\phi, \beta_\phi$ , which is equivalent to the ROC curve of the worst-case MIA, as the *differential privacy distinguisher (DPD) risk* of attack  $\phi(\cdot)$ , following the prior terminology [43]. As DP is a standard notion of privacy leakage in statistical learning, we consider DPD risk an appropriate measure of privacy risk in our settings.

### 3 Record-specific MIA evaluation

In this section, we formalize the traditional and model-seeded privacy games and their estimations of attacker success.

#### 3.1 Traditional privacy game

We refer to the privacy game commonly used in previous work [19, 22, 31, 46, 54] for record-specific evaluation of adversaries under realistic assumptions as the *traditional* game. An attacker's success at inferring a target record's membership is evaluated over multiple runs of the attack, each using a freshly sampled target dataset and the same target record  $x$ . We denote by  $R^T$  resulting risk estimate.

**Definition 2 (Traditional record-specific privacy game).** For target record  $x$ , dataset size  $n$ , training algorithm  $\mathcal{A}(\cdot)$ , and attack  $\phi(\cdot)$ :

1. The challenger samples dataset  $\bar{D} \sim \mathcal{D}^n$  from the distribution with a fresh random seed.
2. The challenger draws a secret bit  $b \in \{0, 1\}$  uniformly at random with a fresh random seed.
3. If  $b = 1$ , the challenger adds target record  $x$  to dataset  $\bar{D}$  to form the target dataset  $D = \bar{D} \cup \{x\}$ . Otherwise,  $D = \bar{D}$ .
4. The challenger trains the target model  $\theta \leftarrow \mathcal{A}(D)$  on dataset  $D$  with a fresh random seed.
5. The adversary outputs a guess  $\hat{b} = \phi(\theta)$ .

### 3.2 Model-seeded privacy game

We now formalize the *model-seeded* privacy game. Here, each run of the game uses the same target record and target dataset, and samples a fresh seed for training the target model. We denote the resulting risk estimate as  $R^{\text{MS}}$ .

**Definition 3 (Model-seeded record-specific privacy game).** *For target record  $x$ , partial dataset  $\bar{D}$ , training algorithm  $\mathcal{A}(\cdot)$ , attack  $\phi(\cdot)$ , and number of runs  $N$ :*

1. *The challenger draws a secret bit  $b \in \{0, 1\}$  uniformly at random with a fresh random seed.*
2. *If  $b = 1$ , the challenger adds target record  $x$  to  $\bar{D}$  to form the target dataset:  $D = \bar{D} \cup \{x\}$ . Otherwise,  $D = \bar{D}$ .*
3. *The challenger trains the target model  $\theta \leftarrow \mathcal{A}(D)$  on dataset  $D$  with a fresh random seed.*
4. *The adversary outputs a guess  $\hat{b} = \phi(\theta)$ .*

In contrast with the traditional privacy game, this game results in an estimate of the target record’s risk within a specific target dataset. By using only the model seed as a source of randomness and eliminating dataset sampling, the ability of the MIA to infer the presence of the target record  $x$  in models trained on  $D$  is evaluated. To the best of our knowledge, this privacy game has so far been used exclusively to evaluate the worst-case attack, where the adversary is assumed to have full knowledge of the target dataset apart from the membership of the target record [3, 17], and never used to evaluate adversaries under realistic assumptions, e.g., with access only to auxiliary data.

### 3.3 The relationship between the games and privacy risk

Consider  $N > 1$  runs of either the model-seeded or traditional game with different random seeds, resulting in a set of guesses  $\{\hat{b}_i\}_{i \in [N]}$  with corresponding secret bits (i.e. membership labels)  $\{b_i\}_{i \in [N]}$ . Let us denote the empirical FPR and FNR obtained in an evaluation using a privacy game for a given attack  $\phi : \Theta \rightarrow [0, 1]$ :

$$\hat{\alpha}_\phi = \frac{\sum_{i=0}^N \mathbb{1}\{\hat{b}_i = 1 \wedge b_i = 0\}}{\sum_{i=0}^N \mathbb{1}\{b_i = 0\}}, \quad \hat{\beta}_\phi = \frac{\sum_{i=0}^N \mathbb{1}\{\hat{b}_i = 0 \wedge b_i = 1\}}{\sum_{i=0}^N \mathbb{1}\{b_i = 1\}} \quad (5)$$

We use  $\hat{\alpha}_\phi^{\text{T}}$  or  $\hat{\alpha}_\phi^{\text{MS}}$  to denote the empirical error rates computed using the traditional (T) and model-seeded (MS) game, respectively, and analogously for  $\hat{\beta}_\phi^{\text{T}}$  and  $\hat{\beta}_\phi^{\text{MS}}$ . We show that, with a sufficiently large number of repetitions of the game with freshly drawn seeds, the empirical FPR and FNR obtained using the model-seeded game converge exponentially fast to the DPD risk as defined in Section 2.3 for any given attack  $\phi$  and record  $x$ :

**Proposition 1 (Model-seeded game converges to DPD risk).** *For any fixed target record  $x$ , partial dataset  $\bar{D} \in \mathbb{D}^{n-1}$ , training algorithm  $T(\cdot)$ , and attack  $\phi(\cdot)$ , we have w.p.  $1 - \rho$  for  $\rho \in (0, 1)$  over  $N$  random coin flips, i.e., fresh seed draws, in the model-seeded game:*

$$|\hat{\alpha}_{\phi}^{MS} - \alpha_{\phi}| \leq \sqrt{\frac{\log(2/\rho)}{2N}}, \quad |\hat{\beta}_{\phi}^{MS} - \beta_{\phi}| \leq \sqrt{\frac{\log(2/\rho)}{2N}} \quad (6)$$

*Proof (Proposition 1).* Consider the set of *in* models  $\{\theta_{\text{in}}^{(i)}\}_{i=1}^N$  and the set of *out* models  $\{\theta_{\text{out}}^{(i)}\}_{i=1}^N$  obtained in the model-seeded game. Let us define  $X_i$  for  $i \in [N]$  as  $X_i = \mathbb{1}[\phi(\theta_{\text{out}}^{(i)}) = 1]$ . The set  $\{X_i\}_{i=1}^N$  is a set of independent Bernoulli random variables. Let  $\bar{X} = \frac{1}{N} \sum_{i=1}^N X_i$ . Then  $\hat{\alpha} = \bar{X}$ . Moreover, we have that  $\alpha = \mathbb{E}[\bar{X}]$ , where the expectations are over sampling  $\{\theta_{\text{out}}^{(i)}\}_{i=1}^N$ . By the Chernoff-Hoeffding inequality, for any  $\gamma > 0$ :

$$\Pr[|\bar{X} - \mathbb{E}[\bar{X}]| \geq \gamma] \leq 2e^{-2N\gamma^2} \quad (7)$$

Thus, with probability at least  $1 - 2e^{-2N\gamma^2}$ :

$$|\bar{X} - \mathbb{E}[\bar{X}]| < \gamma. \quad (8)$$

Setting  $2e^{-2N\gamma^2} = \rho$ , we get:

$$\gamma = \sqrt{\frac{\log(2/\rho)}{2N}}, \quad (9)$$

which yields the sought statement. We get analogous results for  $\beta$ .

In contrast, empirical error rates obtained in the traditional privacy game converge *average* attack success rates over i.i.d. dataset resamples:

**Proposition 2 (Traditional game converges to average privacy risk).** *For any fixed target record  $x$ , dataset size  $n > 1$ , training algorithm  $T(\cdot)$ , and attack  $\phi(\cdot)$ , we have w.p.  $1 - \rho$  for  $\rho \in (0, 1)$  over  $N$  random coin flips, i.e., fresh seed draws, in the traditional game:*

$$|\hat{\alpha}_{\phi}^T - \mathbb{E}_{\bar{D} \sim \mathcal{D}^n} \alpha_{\phi, \bar{D}}| \leq \sqrt{\frac{\log(2/\rho)}{2N}}, \quad |\hat{\beta}_{\phi}^T - \mathbb{E}_{\bar{D} \sim \mathcal{D}^n} \beta_{\phi, \bar{D}}| \leq \sqrt{\frac{\log(2/\rho)}{2N}}, \quad (10)$$

where we explicitly use  $\alpha_{\phi, \bar{D}}$  and  $\beta_{\phi, \bar{D}}$  to emphasize the dependence of  $\alpha_{\phi}$  and  $\beta_{\phi}$  on  $\bar{D}$  in the definition of the hypothesis test in Eq. (2). The proof is analogous to Proposition 1.

Thus, the model-seeded game serves as an estimator of the DPD risk of an attack, as opposed to the traditional game, which estimates an average risk over hypothetical dataset re-samples.

### 3.4 Practical implementation of the privacy games

Algorithm 1 outlines our implementation of the traditional and model-seeded privacy games. In both setups, we construct  $\frac{N_{\text{eval}}}{2} = 500$  ‘in’ and ‘out’ datasets each. In the traditional setup, we sample ‘out’ datasets from the evaluation pool  $D_{\text{eval}}$ , and ensure  $x$  is included in exactly half. In the model-seeded setup, the ‘in’ datasets are equivalent to the full target dataset  $D$ . To maintain an equal dataset size across runs, we construct the ‘out’ datasets by replacing  $x$  with a randomly sampled record  $x_r \sim \text{Unif}[D_{\text{eval}} \setminus D]$ .

---

**Algorithm 1** Practical privacy game implementation
 

---

**Input:** Target record  $x$ , target dataset  $D$ , target training algorithm  $\mathcal{A}(\cdot)$ , evaluation pool  $D_{\text{eval}}$ , partial evaluation pool  $\bar{D}_{\text{eval}} = D_{\text{eval}} \setminus \{x\}$ , MIA  $\phi_x$  with membership score function  $s_x$ , attack thresholds  $\gamma \in \{\gamma_1, \dots, \gamma_m\}$ , number of runs  $N_{\text{eval}}$ , and privacy game flag ( $T$  for traditional,  $MS$  for model-seeded).

**Output:** Empirical error rates  $[\hat{\alpha}_{\phi,1}^{PG}, \dots, \hat{\alpha}_{\phi,m}^{PG}]$  and  $[\hat{\beta}_{\phi,1}^{PG}, \dots, \hat{\beta}_{\phi,m}^{PG}]$  for each attack threshold  $\gamma$ , and summary risk metric  $R^{\text{PG}}$  computed as the ROC AUC of attack  $\phi_x$ .

```

1: for  $i = 0, 1, \dots, \frac{N_{\text{eval}}}{2}$  do
2:   if PG = MS then // model-seeded game
3:      $D_{\text{in}} \leftarrow D$ 
4:     Sample reference record  $x_r \sim \text{Unif}[D_{\text{eval}} \setminus D]$ .
5:     Construct  $D_{\text{out}} \leftarrow D \setminus \{x\} \cup \{x_r\}$ .
6:   else // traditional game
7:     Sample  $\bar{D}_{\text{in}} \sim \text{Unif}[\bar{D}_{\text{eval}}]^{|D|-1}$ 
8:      $D_{\text{in}} \leftarrow \bar{D}_{\text{in}} \cup \{x\}$ 
9:     Sample  $\bar{D}_{\text{out}} \sim \text{Unif}[\bar{D}_{\text{eval}}]^{|D|}$ 
10:   end if
11:   Train evaluation model  $\theta_{\text{in}} = \mathcal{A}(D_{\text{in}})$  with fresh random seed.
12:   Train evaluation model  $\theta_{\text{out}} = \mathcal{A}(D_{\text{out}})$  with fresh random seed.
13:   for  $\gamma_j \in \{\gamma_1, \dots, \gamma_m\}$  do
14:     Compute attack prediction  $\phi_x(\theta_{\text{in}}) = \mathbb{1}[s_x(\theta_{\text{in}}) \geq \gamma_j]$ 
15:      $\hat{b}_{i,j} \leftarrow \phi_x(\theta_{\text{in}})$ 
16:     Compute attack prediction  $\phi_x(\theta_{\text{out}}) = \mathbb{1}[s_x(\theta_{\text{out}}) \geq \gamma_j]$ 
17:      $\hat{b}_{i+\frac{N_{\text{eval}}}{2},j} \leftarrow \phi_x(\theta_{\text{out}})$ 
18:   end for
19:    $b_i \leftarrow 1, b_{i+\frac{N_{\text{eval}}}{2}} \leftarrow 0$ 
20: end for
21: Compute empirical FPR  $\hat{\alpha}_{\phi,j}^{PG}(\{\hat{b}_{i,j}\}_{i=1}^{N_{\text{eval}}}, \{b_i\}_{i=1}^{N_{\text{eval}}})$  for each  $\gamma_j \in [\gamma_1, \dots, \gamma_m]$ 
22: Compute empirical FNR  $\hat{\beta}_{\phi,j}^{PG}(\{\hat{b}_{i,j}\}_{i=1}^{N_{\text{eval}}}, \{b_i\}_{i=1}^{N_{\text{eval}}})$  for each  $\gamma_j \in [\gamma_1, \dots, \gamma_m]$ 
23: Compute summary privacy risk  $R^{\text{PG}} = \text{AUC}(\{\hat{\alpha}_{\phi,j}^{PG}\}_{j=1}^m, \{\hat{\beta}_{\phi,j}^{PG}\}_{j=1}^m)$ 

```

---

## 4 Experimental results

### 4.1 Experimental setup

*Datasets.* We use the Adult [6] and UK Census [36] datasets, commonly tabular datasets used in previous work concerning privacy-preserving synthetic data [19, 31, 46]. Both are de-identified samples of census data containing categorical and continuous demographic features. We partition each dataset into  $D_{\text{aux}}$ , used for MIA development, and  $D_{\text{eval}}$ , used for evaluation. We perform the partitions so that  $|D_{\text{aux}}| = 2 \times |D_{\text{eval}}|$ . We consider  $|D| = 1000$ ,  $D \subset D_{\text{eval}}$ , a common setting in previous work concerning MIAs against synthetic data.

*Target models.* We use Synthpop [35] and Baynet [57] in our main experiments, using the implementations available in the reprosyn [1] repository. We select these generators as they are widely-used, established models.

*MIA methodology.* We use extended-TAPAS, the state-of-the-art query-based attack for SDGs, as originally introduced by Houssiau et al. [22], and extended by Meeus et al. [31]. We train the attack for each target record using auxiliary dataset  $D_{\text{aux}}$  to sample 1000 shadow datasets. TAPAS operates under black-box model access with auxiliary data, but no access to the training data of the target model. We use AUC ROC as a summary metric for privacy risk.

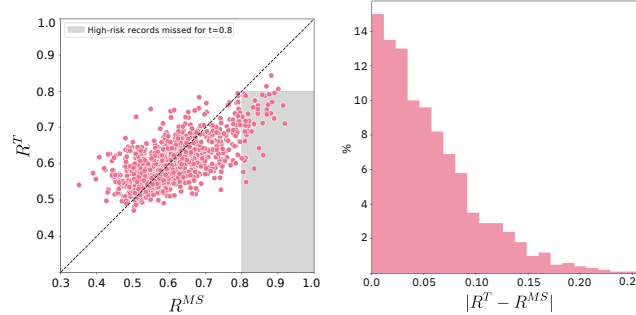
We use the following metrics to compare the traditional and model-seeded risk estimates.

*Miss rate* is the fraction of records classified as high-risk in the model-seeded setup, which are classified as low-risk in the traditional setup. We define a high-risk threshold  $t$  for the MIA AUC, and consider records for which the attack reaches AUC above  $t$  to be high risk. For a subset of records in the target dataset  $S \subseteq D$  and high-risk threshold  $t$ , we compute the miss rate as:  $\text{MR}(S) = \frac{|\{x \in S \mid R^T(x) \leq t \wedge R^{\text{MS}}(x) > t\}|}{|\{x \in S \mid R^{\text{MS}}(x) > t\}|}$ .

*Root Mean Squared Deviation (RMSD)* measures the deviation between traditional and model-seeded risks. For  $S \subseteq D$ , we compute the RMSD between the two risk estimates as  $\text{RMSD}(S, \phi) = \sqrt{\frac{1}{|S|} \sum_{x \in S} (R^T(x) - R^{\text{MS}}(x))^2}$ .

### 4.2 Difference between $R^{\text{MS}}$ and $R^T$

Figure 1 shows the traditional and model-seeded risks to indeed differ substantially. Fig. 1a shows the traditional and model-seeded risks for all 1000 records in  $D$  for the Adult dataset and  $\theta$  Synthpop. This shows that 94% of high-risk records for high-risk threshold  $t = 0.8$  would be incorrectly classified as low-risk when using the traditional setup. Using the traditional setup leads to an RMSD of 0.07, for a value that empirically ranges roughly from 0.5 to 1. Figure 1b shows a histogram of absolute differences between the two risk estimates across records, showing that the estimate would be off by more than 0.1 for 15% of records when using the traditional setup, and could go up to 0.26. Table 1 shows that these



**Fig. 1.** Risk for all 1000 records in  $D$  sampled from the Adult dataset (Synthpop). (a) per-record model-seeded and traditional risks. The shaded area marks all the high-risk records missed in the traditional setup for high-risk threshold  $t = 0.8$ . (b) histogram of per-record absolute differences between the model-seeded and traditional risks.

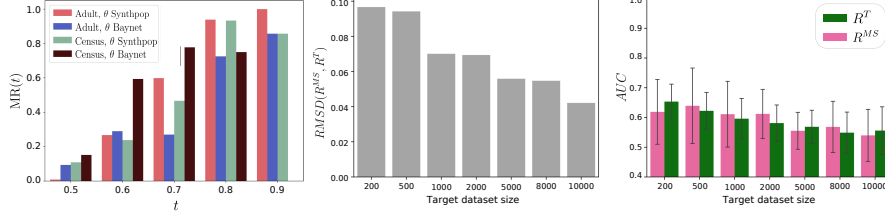
**Table 1.** Miss rate and RMSD across different datasets and target synthetic data generators. We use a high-risk threshold of  $t = 0.8$ .

Dataset	Model	RMSD	MR
Adult	Synthpop	0.07	0.94
	Baynet	0.05	0.73
Census	Synthpop	0.11	0.94
	Baynet	0.04	0.75

results are consistent across setups. The miss rates are consistently high, ranging from 0.73 to 0.94, showing that high-risk records are being incorrectly identified. The majority of the records that are highly vulnerable will thus be incorrectly considered low-risk if MIAs are evaluated using the traditional setup. RMSD ranges from 0.04 to 0.11, a significant error for risk estimated using AUC.

*Different high-risk threshold  $t$  values.* Fig. 2a shows that, for all high-risk thresholds, the miss rates are substantial, reaching values above 20% for all setups for  $t = 0.6$  and up to 80% for  $t = 0.9$ . Using the traditional setup for MIA evaluation thus leads to high-risk records being incorrectly classified as low-risk, regardless of the threshold choice. Notably, we find that the miss rate increases with larger threshold values  $t$ . Identifying high-risk records becomes more difficult as the threshold becomes more strict, and the traditional setup fails to detect an increasing fraction of them.

*Dataset size.* In Section 4.2, we consider target datasets  $D$  of size 1000. We now study how varying  $|D|$  influences the gap between traditional and model-seeded risks. For 20 records from the Adult dataset and with  $\theta$  Synthpop, we compute the risk in both setups for  $|D| \in \{200, \dots, 10000\}$ .

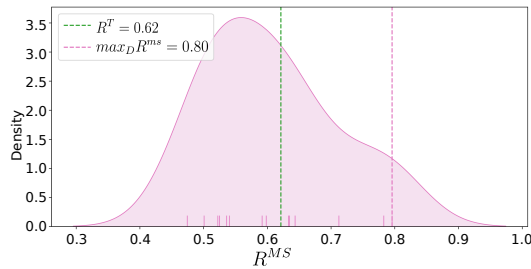


**Fig. 2.** (a) Miss rate for different high-risk thresholds  $t$  for SDG setups. Note that for Census and Baynet, there are no records with  $R^{MS} > 0.9$ , therefore the miss rate is not defined. (b) RMSD between model-seeded and traditional risk per target dataset size. (c) Model-seeded and traditional risk values per target dataset size. For both figures, values are computed across 20 target records.

Fig. 2b shows that the RMSD decreases with dataset size, but remains non-negligible even at  $|D| = 10,000$ . Fig. 2c shows the MIA AUC computed in both setups, averaged across the target records. MIA performance decreases, though it remains better than random, for larger datasets, naturally decreasing the gap. Yet, highly vulnerable records are present even in large datasets, and the two risk estimates do not converge to the same values, showing the importance of using the model-seeded game regardless of dataset size.

### 4.3 Evaluating one record’s risk within different datasets

We use the Adult dataset and the Synthpop model to illustrate an example of the potential negative impact of using the traditional instead of the model-seeded setup. We compute the risk of a single target record in the traditional setup. Then, we compute its model-seeded risk in 15 randomly selected datasets sampled in the traditional setup. As shown in Fig. 3, the model-seeded risk  $R^{MS}$  varies from approximately 0.5 (random guess) to 0.8 (high risk), depending on the dataset. The traditional risk is  $R^T = 0.62$ , underestimating the DPD risk by up to 0.2 in the worst case.



**Fig. 3.** Model-seeded risks of one target record within 15 different datasets and its traditional risk.

## 5 Related work

*Membership inference attacks (MIAs).* Shokri et al. [44] introduced the first MIA against ML models, using model predictions and the *shadow modeling* technique, where multiple models including and excluding the target record are trained to approximate its impact. Various attacks based on shadow modeling have since been proposed [42, 45, 51, 54, 56], typically relying on model loss as the membership signal. The current state-of-the-art attack for ML models, introduced by Carlini et al. [8], uses a likelihood ratio test between loss distributions of models trained with and without the target record.

Tabular synthetic data generators model a dataset as a whole, learning feature distributions and sampling synthetic records [35, 39, 57]. They do not have a notion of per-record loss, rendering standard ML-focused MIAs inapplicable. Instead, specialized record-specific attacks that rely on shadow models and the generated data to assess a record’s influence on synthetic outputs have been proposed [22, 31, 46]. Stadler et al. [46] train meta-classifiers on statistical features from the synthetic data. Houssiau et al. [22] extend this with  $k$ -way queries that count exact matches on random feature subsets, and Meeus et al. [31] further include range-based queries for continuous features.

*Threat models.* A threat model specifies an attacker’s access to the model and data. For synthetic tabular data, most attacks assume black-box access to model outputs [22, 31, 46], though some ML-focused attacks also assume access to predicted probabilities [8, 45, 56] or even labels [11]. White-box attackers have access to model internals, and are more common in vision tasks [4, 12, 20, 30, 38].

Data access defines the data available to the attacker and its relationship to the target data. Attackers are often assumed to access auxiliary datasets drawn from the same distribution as the target [8, 22, 31, 46]. Guépin et al. [19] show this assumption can be relaxed using synthetic data, with performance tradeoffs. Privacy auditing literature typically considers a strong *leave-one-out* adversary with knowledge of all training records except the target [23, 34, 48].

*MIA evaluation.* MIAs are typically evaluated in a *privacy game* between an attacker and a challenger [8, 24, 41, 46, 55]. Ye et al. [54] distinguish between *model-specific* and *record-specific* privacy games. The former evaluates an attacker’s ability to distinguish between records included or excluded from the training data of one model [8, 30, 32, 45, 56], while the latter distinguishes between models trained with and without a specific record, and is standard for evaluating tabular synthetic data attacks [22, 31, 46].

Ye et al. [54] define a privacy game for a *fixed worst-case record and dataset*, typically used to test differential privacy guarantees with a very strong *leave-one-out* attacker [3, 23, 34, 48]. To the best of our knowledge, this privacy game has never been used for weaker attackers or attacks against synthetic data. The model-seeded game is applicable to any attack, regardless of assumptions. The goal of the model-seeded game is to measure the DPD risk for *any* attacker, rather than only the worst-case attacker.



## 6 Discussion & conclusion

We show that the model-seeded privacy game provides an unbiased estimate of a record’s risk, whereas the traditional game averages risk across datasets. Empirically, we show the difference to be significant: the traditional setup results in 85% of high-risk records being misclassified. This confirms that assuming a record’s risk is independent of the dataset is optimistic and can obscure vulnerabilities. Although larger datasets reduce this gap, the model-seeded game consistently offers a more accurate risk estimate and should be preferred.

The exact impact of the dataset on a record’s risk remains open. Prior work suggests that outliers—records with rare or underrepresented features—are more at risk [9, 16, 27, 31, 46]. These characteristics are dataset-specific: a record may be an outlier in one sample but not another, especially in small or high-dimensional data. Larger datasets may better preserve such outlier status, making  $R^{\text{MS}}$  and  $R^{\text{T}}$  more aligned, but not interchangeable.

By formalizing and empirically validating the model-seeded game, we provide a practical and principled tool for assessing privacy risk. We hope this work helps organizations handling sensitive data, such as in healthcare [29] and finance [47], better assess data leakage risks and maintain high privacy standards when releasing synthetic data.

## References

- [1] Alan Turing Institute. 2022. Reprosyn.
- [2] Meenatchi Sundaram Muthu Selva Annamalai, Andrea Gadotti, and Luc Rocher. 2024. A linear reconstruction approach for attribute inference attacks against synthetic data. In *USENIX Security*.
- [3] Meenatchi Sundaram Muthu Selva Annamalai, Georgi Ganey, and Emiliano De Cristofaro. 2024. "What do you want from theory alone?" experimenting with tight auditing of differentially private synthetic data generation. In *USENIX Security*.
- [4] Maryam Azadmanesh, Behrouz Shahgholi Ghahfarokhi, and Maede Ashouri Talouki. 2021. A white-box generator membership inference attack against generative models. In *ISCISC*.
- [5] Borja Balle, Giovanni Cherubin, and Jamie Hayes. 2022. Reconstructing training data with informed adversaries. In *IEEE S&P*.
- [6] Barry Becker and Ronny Kohavi. 1996. Adult. UCI Machine Learning Repository.
- [7] Yuan Cao, Ze Chen, and Zhiyu Quan. [n. d.]. Assessing Insurer’s Litigation Risk: Claim Dispute Prediction with Actionable Interpretations Using Machine Learning Techniques. *SSRN 5126964* ([n. d.]).
- [8] Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. 2022. Membership Inference Attacks From First Principles. In *IEEE S&P*.
- [9] Nicholas Carlini, Matthew Jagielski, Chiyuan Zhang, Nicolas Papernot, Andreas Terzis, and Florian Tramèr. 2022. The privacy onion effect: Memorization is relative. *NeurIPS* (2022).

- 
- [10] Harshdeep Chhikara, Sumit Chhikara, and Lovelesh Gupta. 2025. Predictive Analytics in Finance: Leveraging AI and Machine Learning for Investment Strategies. In *Utilizing AI and Machine Learning in Financial Analysis*. IGI Global Scientific Publishing.
  - [11] Christopher A Choquette-Choo, Florian Tramer, Nicholas Carlini, and Nicolas Papernot. 2021. Label-only membership inference attacks. In *ICML*.
  - [12] Ana-Maria Cretu, Daniel Jones, Yves-Alexandre de Montjoye, and Shruti Tople. 2024. Investigating the Effect of Misalignment on Membership Privacy in the White-box Setting. *PoPETS* (2024).
  - [13] Jinshuo Dong, Aaron Roth, and Weijie J Su. 2022. Gaussian differential privacy. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* (2022).
  - [14] Cynthia Dwork. 2006. Differential Privacy. In *ICALP (Lecture Notes in Computer Science)*.
  - [15] Cynthia Dwork, Aaron Roth, et al. 2014. *The algorithmic foundations of differential privacy*.
  - [16] Vitaly Feldman. 2020. Does learning require memorization? a short tale about a long tail. In *ACM SIGACT Symposium on Theory of Computing*.
  - [17] Georgi Ganey, Meenatchi Sundaram Muthu Selva Annamalai, and Emiliano De Cristofaro. 2025. The Elusive Pursuit of Reproducing PATE-GAN: Benchmarking, Auditing, Debugging. *TMLR* (2025).
  - [18] Vincent Guan, Florent Guépin, Ana-Maria Cretu, and Yves-Alexandre de Montjoye. 2024. A zero auxiliary knowledge membership inference attack on aggregate location data. *PoPETS* (2024).
  - [19] Florent Guépin, Matthieu Meeus, Ana-Maria Cretu, and Yves-Alexandre de Montjoye. 2023. Synthetic is all you need: removing the auxiliary data assumption for membership inference attacks against synthetic data. In *ESORICS*.
  - [20] Jamie Hayes, Luca Melis, George Danezis, and Emiliano De Cristofaro. 2019. LOGAN: Membership Inference Attacks Against Generative Models. *PoPETS* (2019).
  - [21] Zecheng He, Tianwei Zhang, and Ruby B. Lee. 2019. Model inversion attacks against collaborative inference. In *Computer Security Applications Conference*.
  - [22] Florimond Houssiau, James Jordon, Samuel N Cohen, Owen Daniel, Andrew Elliott, James Geddes, Callum Mole, Camila Rangel-Smith, and Lukasz Szpruch. 2022. Tapas: a toolbox for adversarial privacy auditing of synthetic data. (2022).
  - [23] Matthew Jagielski, Jonathan Ullman, and Alina Oprea. 2020. Auditing Differentially Private Machine Learning: How Private is Private SGD?. In *NeurIPS*.
  - [24] Bargav Jayaraman, Lingxiao Wang, Katherine Knipmeyer, Quanquan Gu, and David Evans. 2021. Revisiting Membership Inference Under Realistic Assumptions. *PoPETS 2021* (2021).
  - [25] Peter Kairouz, Sewoong Oh, and Pramod Viswanath. 2015. The composition theorem for differential privacy. In *ICML*.
  - [26] Bogdan Kulynych, Juan Felipe Gomez, Georgios Kaissis, Flavio Calmon, and Carmela Troncoso. 2024. Attack-Aware Noise Calibration for Differential Privacy. In *NeurIPS*.
  - [27] Bogdan Kulynych, Mohammad Yaghini, Giovanni Cherubin, Michael Veale, and Carmela Troncoso. 2022. Disparate Vulnerability to Membership Inference Attacks. *PoPETS* (2022).
  - [28] Sasi Kumar and Reza Shokri. 2020. ML Privacy Meter: Aiding regulatory compliance by quantifying the privacy risks of machine learning. In *Workshop on Hot Topics in Privacy Enhancing Technologies (HotPETs)*.

- 
- [29] Eyal Lotan, Charlotte Tschider, Daniel K Sodickson, Arthur L Caplan, Mary Bruno, Ben Zhang, and Yvonne W Lui. 2020. Medical imaging and privacy in the era of artificial intelligence: myth, fallacy, and the future. *Journal of the American College of Radiology* (2020).
  - [30] Tomoya Matsumoto, Takayuki Miura, and Naoto Yanai. 2023. Membership inference attacks against diffusion models. In *IEEE S&P Workshops (SPW)*.
  - [31] Matthieu Meeus, Florent Guepin, Ana-Maria Crețu, and Yves-Alexandre de Montjoye. 2023. Achilles’ heels: vulnerable record identification in synthetic data publishing. In *ESORICS*.
  - [32] Matthieu Meeus, Lukas Wutschitz, Santiago Zanella-Béguelin, Shruti Tople, and Reza Shokri. 2025. The Canary’s Echo: Auditing Privacy Risks of LLM-Generated Synthetic Text.
  - [33] Milad Nasr, Reza Shokri, and Amir Houmansadr. 2019. Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In *IEEE S&P*.
  - [34] Milad Nasr, Shuang Song, Abhradeep Thakurta, Nicolas Papernot, and Nicholas Carlini. 2021. Adversary Instantiation: Lower Bounds for Differentially Private Machine Learning. In *IEEE S&P*.
  - [35] Beata Nowok, Gillian M. Raab, and Chris Dibben. 2016. synthpop: Bespoke Creation of Synthetic Data in R. *Journal of Statistical Software* (2016).
  - [36] Office for National Statistics. 2011. Census Microdata Teaching Files.
  - [37] Richard Osuala, Daniel M. Lang, Anneliese Riess, et al. 2025. Enhancing the Utility of Privacy-Preserving Cancer Classification Using Synthetic Data. In *Artificial Intelligence and Imaging for Diagnostic and Treatment Challenges in Breast Care*. Springer Nature Switzerland.
  - [38] Yan Pang, Tianhao Wang, Xuhui Kang, Mengdi Huai, and Yang Zhang. 2025. White-box Membership Inference Attacks against Diffusion Models.
  - [39] Haoyue Ping, Julia Stoyanovich, and Bill Howe. 2017. DataSynthesizer: Privacy-Preserving Synthetic Datasets (*SSDBM*). ACM.
  - [40] Joseph Pollock, Igor Shilov, Euodia Dodd, and Yves-Alexandre de Montjoye. 2024. Free Record-Level Privacy Risk Evaluation Through Artifact-Based Methods. *arXiv preprint arXiv:2411.05743* (2024).
  - [41] Apostolos Pyrgelis, Carmela Troncoso, and Emiliano De Cristofaro. 2018. Knock Knock, Who’s There? Membership Inference on Aggregate Location Data. (2018).
  - [42] Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, Yann Ollivier, and Hervé Jégou. 2019. White-box vs black-box: Bayes optimal strategies for membership inference. In *ICML*.
  - [43] Ahmed Salem, Giovanni Cherubin, David Evans, Boris Köpf, Andrew Paverd, Anshuman Suri, Shruti Tople, and Santiago Zanella-Béguelin. 2023. SoK: Let the privacy games begin! A unified treatment of data inference privacy in machine learning. In *IEEE S&P*.
  - [44] Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. 2017. Membership inference attacks against machine learning models. In *IEEE S&P*.
  - [45] Liwei Song and Prateek Mittal. 2021. Systematic evaluation of privacy risks of machine learning models. In *USENIX Security*.
  - [46] Theresa Stadler, Bristena Oprisanu, and Carmela Troncoso. 2022. Synthetic data-anonymisation groundhog day. In *USENIX Security*.
  - [47] Synthetic Data Expert Group, Financial Conduct Authority. 2024. Report: Using Synthetic Data in Financial Services.

- 
- [48] Florian Tramer, Andreas Terzis, Thomas Steinke, Shuang Song, Matthew Jagielski, and Nicholas Carlini. 2022. Debugging Differential Privacy: A Case Study for Privacy Auditing.
  - [49] Zhibo Wang, Mengkai Song, Zhifei Zhang, Yang Song, Qian Wang, and Hairong Qi. 2019. Beyond inferring class representatives: User-level privacy leakage from federated learning. In *IEEE INFOCOM 2019-IEEE conference on computer communications*.
  - [50] Larry Wasserman and Shuheng Zhou. 2010. A statistical framework for differential privacy. *J. Amer. Statist. Assoc.* (2010).
  - [51] Lauren Watson, Chuan Guo, Graham Cormode, and Alexandre Sablayrolles. 2021. On the Importance of Difficulty Calibration in Membership Inference Attacks. In *International Conference on Learning Representations*.
  - [52] Lei Xu, Maria Skoularidou, Alfredo Cuesta-Infante, and Kalyan Veeramachaneni. 2019. Modeling tabular data using conditional gan. *NeurIPS* (2019).
  - [53] Ziqi Yang, Jiyi Zhang, Ee-Chien Chang, and Zhenkai Liang. 2019. Neural network inversion in adversarial setting via background knowledge alignment. In *ACM CCS*.
  - [54] Jiayuan Ye, Aadyaa Maddi, Sasi Kumar Murakonda, Vincent Bindschaedler, and Reza Shokri. 2022. Enhanced Membership Inference Attacks against Machine Learning Models. In *ACM CCS*.
  - [55] S. Yeom, I. Giacomelli, M. Fredrikson, and S. Jha. 2018. Privacy Risk in Machine Learning: Analyzing the Connection to Overfitting. In *IEEE Computer Security Foundations Symposium*.
  - [56] Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. 2024. Low-Cost High-Power Membership Inference Attacks. In *ICML*.
  - [57] Jun Zhang, Graham Cormode, Cecilia M. Procopiuc, Divesh Srivastava, and Xiaokui Xiao. 2017. PrivBayes: Private Data Release via Bayesian Networks. (2017).

# Membership Inference Attacks Beyond Overfitting

Mona Khalil<sup>1</sup> (✉), Alberto Blanco-Justicia<sup>1</sup>, Najeeb Jebreel<sup>1</sup>, and Josep Domingo-Ferrer<sup>1,2</sup>

<sup>1</sup> Universitat Rovira i Virgili,  
Department of Computer Engineering and Mathematics,  
CYBERCAT-Center for Cybersecurity Research of Catalonia,  
Av. Països Catalans 26, 43007 Tarragona, Catalonia  
{mona.khalil, alberto.blanco, najeeb.jebreel, josep.domingo}@urv.cat  
<sup>2</sup> LAAS-CNRS, Université de Toulouse, 7 Av. du Colonel Roche, 31400 Toulouse,  
France

**Abstract.** Membership inference attacks (MIAs) against machine learning (ML) models aim to determine whether a given data point was part of the model training data. These attacks may pose significant privacy risks to individuals whose *sensitive* data were used for training, which motivates the use of defenses such as differential privacy, often at the cost of high accuracy losses. MIAs exploit the differences in the behavior of a model when making predictions on samples it has seen during training (*members*) versus those it has not seen (*non-members*). Several studies have pointed out that model overfitting is the major factor contributing to these differences in behavior and, consequently, to the success of MIAs. However, the literature also shows that even non-overfitted ML models can leak information about a small subset of their training data. In this paper, we investigate the root causes of membership inference vulnerabilities beyond traditional overfitting concerns and suggest targeted defenses. We empirically analyze the characteristics of the training data samples vulnerable to MIAs in models that are not overfitted (and hence able to generalize). Our findings reveal that these samples are often outliers within their classes (*e.g.*, noisy or hard to classify). We then propose potential defensive strategies to protect these vulnerable samples and enhance the privacy-preserving capabilities of ML models. Our code is available at [https://github.com/najeebjebreel/mia\\_analysis](https://github.com/najeebjebreel/mia_analysis).

**Keywords:** Machine learning · Privacy · Membership inference attacks.

## 1 Introduction

Machine learning (ML) has demonstrated remarkable performance across a wide range of tasks [15,9,27]. This success is mainly attributed to the availability of large and diverse data for training, along with advances in learning algorithms and computational capabilities.

However, training data often contain sensitive information related to individuals, such as personal photos [20], confidential texts [6], clinical records [21], and financial details [30]. Unauthorized access to or leakage of such data can lead to significant privacy risks and adverse consequences for affected individuals.

Trained ML models can memorize and inadvertently reveal sensitive information about their training data [40,5,48], making them vulnerable to several privacy attacks, such as extraction attacks [6], property inference attacks [13], and membership inference attacks (MIAs) [38].

MIAs [38,35,47], the focus of this paper, aim to determine whether a specific data point was part of the training data of a given model. Although they may not seem dangerous at first glance, they can pose serious privacy risks to individuals in specific scenarios. For example, knowing that a specific patient’s clinical record was used to train a model associated with a sensitive disease can reveal with high confidence that the patient suffers from this disease.

Several studies have demonstrated a strong connection between training data memorization and the phenomenon of overfitting [47,5,46]. Overfitting occurs when a model not only learns general patterns, but also captures sample-specific details and noise, which leads to a noticeable difference in its behavior in training data (*members*) compared to unseen data (*non-members*) [38,47,16,46]. MIAs leverage this differential behavior [38,29,41,4].

Various defenses against MIAs have been proposed and can be categorized into certified and practical defenses. Certified defenses provide formal privacy guarantees through differential privacy (DP) [1], but often result in reduced model utility and high computational costs. Practical defenses, on the other hand, offer empirical privacy protection with the goal of maintaining the utility of the model [43,28,19,41,2]. These practical defenses primarily aim to mitigate overfitting and develop models with better generalization capabilities, thus reducing the effectiveness of MIAs while preserving utility. However, even models designed to generalize well can inadvertently leak information about a small portion of the training data, making them vulnerable to MIAs [25,4].

**Contributions:** In this paper, we address two key questions: *Q1: What makes certain samples vulnerable to MIAs even in non-overfitted models?* and *Q2: How can these samples be effectively protected?*

To answer these questions, we performed experiments on various data sets and models to identify factors that contribute to the vulnerability of MIA beyond overfitting. We systematically characterize what makes samples vulnerable through visual analysis, feature-space geometry, and model explanation techniques. We find that outliers—samples that are far from their class centroid—are particularly vulnerable. We then suggest and discuss potential defensive strategies to protect these vulnerable samples and thereby enhance privacy.

The remainder of this paper is organized as follows. Section 2 provides background on ML overfitting and differential privacy. Section 3 discusses related work on membership inference attacks and defenses, and factors that contribute to the success of MIAs. Section 4 describes the data sets, models, and experimental setup. Section 5 empirically investigates the causes of MIA beyond over-

fitting and discusses the results obtained. Section 6 discusses potential solutions for protecting vulnerable samples. Section 7 summarizes our findings and suggests future research directions. Additional experimental details are provided in the [supplementary materials](#).

## 2 Background

### 2.1 Machine Learning Overfitting

In this paper, we focus on predictive deep neural network (DNNs) utilized as  $m$ -class classifiers, with the cross-entropy (CE) loss:

$$\mathcal{L}(F_\theta, z) = - \sum_{i=0}^{m-1} y_i \log(F_\theta(x)_i), \quad (1)$$

where  $x$  are the input features,  $y_i$  is the one-hot encoded label vector, and  $F_\theta(x)_i$  is the predicted probability for class  $i$ .

One of the potential problems of ML training is overfitting. Overfitting is an undesirable training outcome in which the model fits too closely to the training data but performs poorly on the test data, resulting in a high generalization error [23]. Overfitting can arise from various factors, including overparameterized models, insufficient training data, high data dimensionality, or suboptimal hyperparameter selection (*e.g.* batch size, learning rate). In addition, [8] highlight frequent data exposure during training and sharp loss functions as factors that exacerbate MIA risks. In particular, [12] demonstrate that some degree of memorization may be essential for optimal generalization, particularly when learning from rare or unique instances.

Since best practices of ML emphasize avoiding overfitting to enhance generalization and maximize utility, our work focuses on identifying training samples that remain vulnerable to MIAs even in non-overfitted models.

### 2.2 Differential Privacy (DP)

Differential privacy (DP [11]) ensures that the inclusion or exclusion of a single data point in a data set does not significantly affect the output of a statistical function. Formally, a mechanism  $M$  satisfies  $(\epsilon, \delta)$ -DP if, for any two neighboring data sets  $D$  and  $D'$  (differing by one data point) and any subset  $S$  of outcomes:

$$\Pr[M(D) \in S] \leq e^\epsilon \Pr[M(D') \in S] + \delta, \quad (2)$$

where  $\epsilon$  is the privacy budget (smaller values imply stronger privacy), and  $\delta$  is the probability of exceeding the budget.

In DNN training, DP is typically implemented via DP-SGD [1], which clips per-example gradients to bound sensitivity and adds Gaussian noise to the batch gradient during training. However, DP-SGD introduces challenges, including complex hyperparameter tuning, increased training time, and reduced model utility [32,2].

### 3 Related Work

#### 3.1 Black-box MIA Approaches

We focus on black-box MIAs since, on the one hand, according to [34] they are (or can be) as good as any white-box MIAs. There are several approaches to conducting black-box MIAs, each leveraging different aspects of the model output to distinguish between members and non-members. Shadow model attacks [38] train multiple models to mimic the target model behavior, and train an ML attack model on the predictions of the shadow models to distinguish members from non-members. [47] infer a sample as a member if its loss is less than the average training loss. [35] threshold the confidence score of a sample to infer membership, with higher confidence indicating membership. [41] utilize prediction entropy, with lower entropy indicating membership. The likelihood ratio attack (LiRA) of [4] applies hypothesis testing using Gaussian distributions fitted to the output of multiple models (trained with and without the target samples), achieving more reliable detection, but requiring extensive computation.

#### 3.2 Defenses Against Membership Inference Attacks

To mitigate membership inference attacks (MIAs), various defenses have been proposed. Differential privacy (DP) methods, such as DP-SGD (noise-added gradient descent) [1] and PATE (ensemble training with noisy voting) [31], provide formal privacy guarantees, but often reduce model utility and increase computational costs [38,47,33,18].

Anti-overfitting strategies, which maintain better utility while mitigating MIAs, include early stopping [7,41] and regularization techniques: L2 regularization penalizes large parameters; dropout randomly deactivates units during training [42,35]; adversarial regularization [28] modifies the loss function; and label smoothing replaces hard labels with soft distributions [43].

Output masking defenses restrict prediction details by releasing only top-k probabilities or class labels [38], though top-k leakage remains a limitation. MemGuard [19] further perturbs confidence scores to confuse attackers. Knowledge distillation methods, such as DMP (low-entropy training) [37] and SELENA (sub-model distillation) [44], transfer knowledge from teacher to student models to enhance privacy.

#### 3.3 Understanding MIA Vulnerabilities Beyond Overfitting

While overfitting is a known primary cause of MIA vulnerabilities [47], privacy leakage may also occur in non-overfitted models [25,4]. [25] identify vulnerable samples in well-generalized models as those with few neighbors in the intermediate feature space. [4] use shadow model training to model per-example loss distributions for members and non-members as Gaussian distributions and detect members via likelihood ratio tests. [24] analyze loss trajectories during training to identify vulnerable samples.



Our work differs from these studies in two key ways: (1) We provide a comprehensive visual and geometric analysis of vulnerable samples using t-SNE visualizations [26], combined with model explanation techniques (Grad-CAM [36]) to reveal why specific samples are vulnerable. We show that models tend to focus on non-relevant features for outlier samples relative to their class centroids. (2) Whereas prior work [25,4,24] primarily informs attack design, we leverage our analysis to suggest suitable defenses and propose a novel logit-reweighting method specifically targeting geometrically identified vulnerable samples.

## 4 Experimental Setup

**Data sets and models.** We used two benchmark data sets commonly used in the literature of MIAs, namely *Purchase100* [38] and *CIFAR-10* [22]. For *Purchase100*, we used a fully connected network (FCN) as in [37]. For *CIFAR-10*, we employed two convolutional neural network architectures: DenseNet-12 [17] and ResNet-18 [15]. The utility of the model was measured through accuracy.

**Attacks and defenses.** We evaluated two black-box MIAs (loss-based [47], entropy-based [35]) using AUC and the attacker’s advantage [47]. MIA AUC measures the overall attack performance across all decision thresholds using the Area Under the ROC Curve. An AUC of 50% indicates random guessing (perfect privacy), while higher values indicate more effective attacks and greater privacy leakage. MIA attacker’s advantage is defined as  $2 \cdot \Pr[\text{correct guess}] - 1$  [47], which is equivalent to  $\max_{\tau}(\text{TPR}(\tau) - \text{FPR}(\tau))$  across all decision thresholds  $\tau$ . An advantage of 0% means no benefit over random guessing, while higher percentages indicate greater privacy violations.

In addition, we identified the most vulnerable samples as true positive samples (TP) at a low false positive rate (FPR), as suggested in [4]; these are the samples that are the most reliably detectable by the attacker. We considered the following defenses: early stopping [41], L2-regularization [38], regularization and dropout (RegDrop) [2], label smoothing (LS) [43], and DP-SGD [1].

More details on system specifications, data set descriptions, models, attacks, defenses, and training settings are provided in [supplementary materials](#).

## 5 Results and Discussion

In this section, we first study the impact of overfitting on model utility and MIAs. Then, we apply a set of representative defenses against MIAs (described in Section 4) and analyze their impact on the utility of the model and MIAs. After that, we analyze why some training samples of non-overfitted models are still vulnerable to MIAs.

### 5.1 Impact of Overfitting

Overfitting has an impact on the utility of the model and the effectiveness of MIAs. Let us examine this impact in depth.

**Separation between members and non-members.** Figure 1 illustrates the histograms of the distributions of scaled logits [4] for member and non-member data points across different epochs during the training of the CIFAR10-DenseNet-12. The figure also displays metrics related to model utility, namely training accuracy (Train Acc) and test accuracy (Test Acc), as well as metrics related to membership inference attacks (MIA), specifically MIA AUC and MIA attacker advantage (MIA Adv). These metrics indicate that as the model trains and begins to overfit, the gap between training and test accuracy increases, and the separation between member and non-member data points becomes more pronounced, thereby increasing the model’s vulnerability to MIAs.

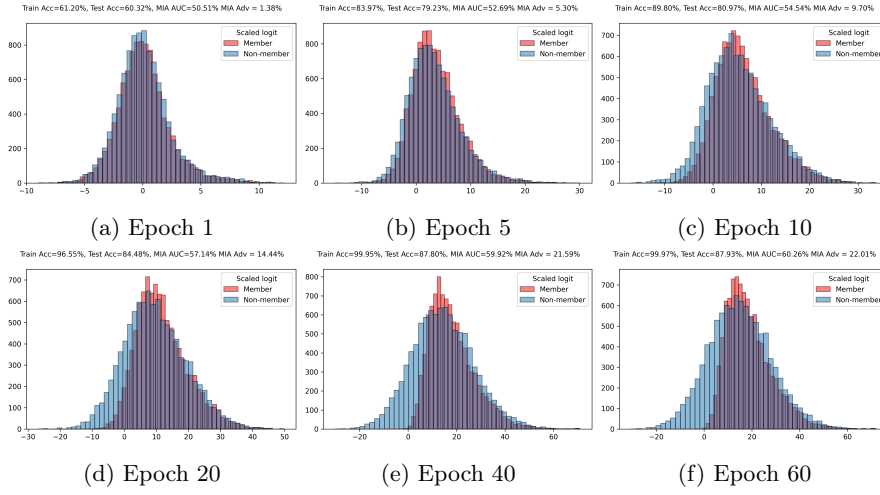


Fig. 1: Impact of overfitting in CIFAR10-DenseNet. Distributions of scaled logits for member and non-member data points, accuracy metrics, and MIA metrics for several epochs.

**Overfitting and model complexity.** Table 1 compares accuracy and MIA metrics for two models whose number of parameters are significantly different. It can be seen that in the larger model the gap between training and test performance is greater, which makes MIAs more effective. This is a sign of overfitting by the larger model.

## 5.2 Effectiveness of Defenses Against MIAs

This section evaluates several defense mechanisms (described in Section 3.2) designed to mitigate the vulnerability of DNN models to membership inference attacks. These defenses are tested on two benchmarks: Purchase100-FCN and CIFAR10-DenseNet-12. The performance of these defenses is evaluated in terms of utility, runtime, and resistance to MIAs.

Table 1: Impact of model complexity

Method	# params	Train Acc	Test Acc	MIA AUC	MIA Adv.
DenseNet	~770,000	99.97	87.91	60.27	22.07
ResNet	~11,170,000	99.26	82.79	64.45	28.31

An ideal defense should maintain or exceed the model’s original accuracy (due to improved generalization) with a similar or lower runtime. In terms of privacy protection, an optimal defense should render MIAs as ineffective as random guessing, achieving an AUC of 50% and a zero advantage in predicting membership status.

Table 2: Performance of defenses with Purchase100-FCN. Best figures are bold-faced, second-best are underlined.

Method	Train Acc (%)	Test Acc (%)	Runtime (s)	MIA AUC (%)	MIA Adv. (%)
Original	<u>97.76</u>	87.54	<u>1201</u>	57.27	13.86
Early stopping	<u>96.88</u>	<b>89.58</b>	<b>200</b>	55.07	10.40
Regularization( $\lambda=5e-4$ )	94.91	89.34	1209	53.25	7.26
Regularization( $\lambda=1e-3$ )	92.63	<u>88.37</u>	1205	52.22	4.87
Regularization( $\lambda=5e-3$ )	77.76	76.16	1207	<u>50.92</u>	<u>1.73</u>
RegDrop( $\lambda=5e-4, dr=0.25$ )	90.02	87.14	1489	<u>51.87</u>	<u>3.70</u>
RegDrop( $\lambda=5e-4, dr=0.50$ )	86.52	84.45	1320	51.44	2.46
Label smoothing	<b>99.15</b>	88.52	1699	59.43	16.43
DP( $\epsilon = 2.38$ )	61.71	61.21	3507	<b>50.36</b>	<b>0.70</b>

Table 2 shows the performance of defenses with the Purchase100-FCN benchmark. The original model achieved a high training accuracy of 97.76% and a test accuracy of 87.54%. However, it showed vulnerability to MIAs with MIA AUC 57.27% and MIA advantage 13.86%.

Early stopping achieved the best test accuracy to 89.58% and the shortest runtime (200 seconds). It also slightly decreased the MIA AUC and advantage to 55.07% and 10.40%, respectively. This is because early stopping in this benchmark managed to stop the model training process before seriously overfitting the training data.

Regularization with different  $\lambda$  values showed a trend of degrading accuracy and improving privacy as the regularization strength increased. Regularization with  $\lambda = 5e-4$  improved test accuracy to 89.34%, reduced MIA AUC to 53.25%, and MIA advantage to 7.26%. Increasing  $\lambda$  to  $1e-3$  further reduced the MIA AUC and advantage to 52.22% and 4.87%, respectively, with a slight drop in test accuracy to 88.37%. The highest regularization ( $\lambda = 5e-3$ ) significantly reduced both training and test accuracy (77.76% and 76.16%), but achieved the lowest MIA AUC (50.92%) and MIA advantage (1.73%). This indicates a strong trade-off between model performance and privacy, where higher regularization

reduces overfitting and enhances privacy at the cost of accuracy. Regularization also took a runtime similar to that of the original training. These results suggest that regularization with  $\lambda = 1e - 3$  struck the best balance between utility, runtime and privacy for this benchmark.

RegDrop with  $\lambda = 5e - 4$  and dropout rates 0.25 and 0.50 slightly degraded utility, but significantly reduced MIA effectiveness. For dropout rate 0.25 we obtained test accuracy 87.14%, MIA AUC 51.87%, and MIA advantage 3.70%. Increasing the dropout rate to 0.50 reduced test accuracy to 84.45% but further lowered the MIA AUC to 51.44% and the MIA advantage to 2.46%. These results suggest that RegDrop offered the best balance between utility and privacy for this benchmark.

Label smoothing achieved relatively high test accuracy (88.52%). However, it increased the model susceptibility to MIAs, as reflected by the MIA AUC of 59.43% and advantage of 16.43%. This result indicates that, while label smoothing increases training accuracy, it may cause the model to leave a distinguishable pattern in predictions of training samples, thus exacerbating the vulnerability to MIAs.

Differential privacy with  $\epsilon = 2.38$  drastically reduced the MIA AUC to 50.36% and MIA advantage to 0.70%, offering the strongest defense against MIAs. However, this came at the expense of model utility, because training and test accuracy dropped to 61.71% and 61.21%, respectively. The significant accuracy reduction highlights the trade-off of DP between strong privacy guarantees and model utility. The runtime (3507 seconds) was also the highest, indicating a substantial computational cost to reach convergence when training under DP.

In summary, we can see diverse trade-offs between model utility, computational cost, and privacy among defenses. Early stopping provided the best balance between utility and runtime. However, it only slightly mitigated MIAs. *Moderate regularization showed the best utility-runtime-privacy trade-off among all defenses for this benchmark.* RegDrop, particularly at a low dropout rate, offered the best balance between utility and privacy. It achieved privacy protection close to that of DP with much better utility and runtime. Although differential privacy provided the strongest privacy protection, this came at the cost of significant accuracy loss and increased runtime. An interesting note is that regularization with  $\lambda = 5e - 3$  achieved an effectiveness against MIAs close to that of DP but with much better accuracy and runtime. Label smoothing, despite its high training accuracy, increased MIA vulnerability, suggesting that its application requires careful tuning.

Table 3 reports the same defense analysis for the CIFAR10-DenseNet-12 benchmark. The results show that the original model achieved an extremely high training accuracy (99.97%) and a test accuracy 87.91%. However, the high MIA AUC (60.27%) and advantage (22.07%) indicate overfitting, making the model vulnerable to MIAs.

Early stopping maintained a high training accuracy (99.97%) and slightly improved the test accuracy to 87.93%. It also reduced the runtime significantly to 2024 seconds. However, resistance to MIAs was not improved.

Table 3: Performance of defenses with CIFAR10-DenseNet-12. Best figures are boldfaced, second-best are underlined.

Method	Train Acc (%)	Test Acc (%)	Runtime (s)	MIA AUC (%)	MIA Adv. (%)
Original	<u>99.97</u>	87.91	<u>3558</u>	60.27	22.07
Early stopping	<u>99.97</u>	87.93	<b>2024</b>	60.21	21.96
Regularization( $\lambda=5e-4$ )	<b>99.99</b>	<u>91.46</u>	3573	57.11	19.13
Regularization( $\lambda=1e-3$ )	99.95	89.61	3564	58.07	20.52
Regularization( $\lambda=5e-3$ )	60.35	59.71	3574	<u>50.06</u>	<u>0.82</u>
RegDrop( $\lambda=5e-4$ ,dr=0.25)	99.85	<b>91.78</b>	3616	56.00	15.44
RegDrop( $\lambda=5e-4$ ,dr=0.50)	91.97	84.89	3610	53.61	7.52
Label smoothing	<b>99.99</b>	86.47	3539	67.33	37.04
DP( $\epsilon = 4.95$ )	59.51	59.55	7738	<b>50.00</b>	<b>0.53</b>

Regularization with  $\lambda = 5e - 4$  achieved the second highest test accuracy (91.46%). It also slightly improved privacy with MIA AUC 57.11% and advantage 19.13%. Increasing regularization to  $\lambda = 1e - 3$  slightly improved test accuracy (89.61%) and privacy metrics (MIA AUC 58.07% and advantage 20.52%). At the highest regularization strength ( $\lambda = 5e - 3$ ), there was a drastic drop in both training and test accuracy (60.35% and 59.71%, respectively), but this setting achieved the lowest MIA AUC (50.06%) and advantage (0.82%), indicating strong privacy protection at the cost of performance.

RegDrop with  $\lambda = 5e - 4$  and a dropout rate of 0.25 achieved the best balance, with the highest test accuracy (91.78%) and improved privacy metrics (MIA AUC 56.00% and advantage 15.44%). Increasing the dropout rate to 0.50 reduced test accuracy to 84.89% but further enhanced privacy (MIA AUC 53.61% and advantage 7.52%). This demonstrates RegDrop’s effectiveness in mitigating overfitting and enhancing privacy while maintaining reasonable accuracy.

Label smoothing achieved the highest training accuracy (99.99%) but offered a lower test accuracy (86.47%) compared to the baseline original model. This method actually increased the model’s vulnerability to MIAs, with the highest MIA AUC (67.33%) and MIA advantage (37.04%). This suggests that, whereas label smoothing can improve training performance, it may render the model more susceptible to privacy attacks.

Differential privacy with  $\epsilon = 4.95$  provided the strongest defense against MIAs, achieving the lowest MIA AUC (50.00%) and advantage (0.53%), almost similar to random guessing. However, this came with significant reductions in both training and test accuracies (59.51% and 59.55%, respectively) and a substantial runtime (7738 seconds).

In summary, regularization and its combination with dropout achieved the best utility-privacy balance. Early stopping offered computational efficiency, but weak privacy protection. Differential privacy provided strong formal guarantees, but degraded performance and increased computational costs. Label smoothing improved training accuracy but increased MIA vulnerability. These findings suggest that selecting appropriate regularization and dropout parameters is the

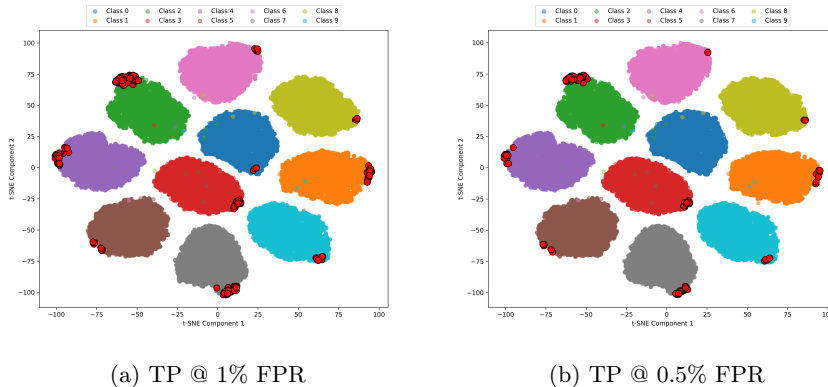


Fig. 2: t-SNE visualization of vulnerable samples (circled red) w.r.t their class samples

most effective approach to balancing utility, runtime, and privacy when training DNN models, as also observed by [2].

Despite its strong generalization capabilities, the best-performing CIFAR10-DenseNet-12 model (RegDrop with  $\lambda = 5e - 4$  and a dropout rate 0.25) still exhibited an MIA AUC 56.00% and an MIA advantage 15.44%, which are both above the level expected from random guessing. This raises the crucial question addressed in this paper: **Why do membership inference attacks perform better than random guessing on models exhibiting good generalization, and what characteristics define the training samples that remain vulnerable to MIAs?** To answer this question, the following section provides a thorough examination of the training samples that continue to be susceptible to MIAs even after successfully mitigating overfitting in the CIFAR10-DenseNet-12 model (RegDrop with  $\lambda = 5e - 4$  and a dropout rate of 0.25).

### 5.3 Vulnerable Samples Beyond Overfitting

We focus on the most vulnerable training samples, selecting true positives (TP) with a 1% false positive rate (FPR) following [4]. We directly used the loss values of the train and test samples from the target model.

The t-SNE visualization of the latent features of these samples in Figure 2a shows that these vulnerable samples are located primarily on the borders of their respective class clusters. This suggests that these samples differ significantly from the majority, likely being hard-to-classify, noisy, or outliers. Such characteristics may cause the model to memorize these samples based on specific details rather than relevant class patterns, leading to overconfidence in predictions and increased vulnerability to MIAs. True positives (TP) with a false positive rate of 0.5% (FPR) are also shown in Figure 2b.

To further investigate the nature of these boundary samples that remain vulnerable to MIAs, we analyze their characteristics compared to typical class samples. For CIFAR10-DenseNet-12, Figure 3 provides visualizations and explanations of samples close to the class centroid and vulnerable samples. Explanations are based on the Grad-CAM method (gradient-weighted class activation mapping, [36]), which highlights the pixels responsible for decisions. Specifically:

- Figure 3a shows the inlier images close to their centroids (first row) and the images most vulnerable to MIAs from each class (second row). We can see that inlier samples are clear and easy to classify while the most vulnerable samples are noisy (*e.g.*, the cat hidden by the red net), unclear (*e.g.*, the tiny bird in the blue sky and the man riding the horse), or hard to classify (*e.g.*, the black cat and the big face frog).
- Figure 3b gives the Grad-CAM explanations of the classification decisions for the images in Figure 3a. In the case of the inlier images, the relevant pixels corresponding to the class’s general patterns were identified. For the vulnerable examples, non-relevant pixels were generally identified. In most cases, these identified pixels were related to noise details (*e.g.*, the red net obscuring the cat) or sample-specific details (*e.g.*, the rear traffic light of the car and the people riding the truck).

These observations indicate that noisy or unclear samples may inherently resist MIAs because they do not facilitate the clear identification of individual data points. In contrast, clear samples with unique or untypical features — those that are difficult to classify — are particularly vulnerable to MIAs. Even in a model with good generalization capability, overfitting to these unique aspects can lead to memorization, which attackers can exploit.

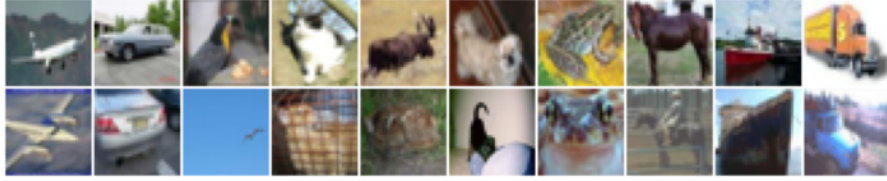
## 6 Potential Solutions

This section explores potential defenses to mitigate the memorization of vulnerable samples in DNNs, depending on whether these samples are identified beforehand. Most defenses are based on established techniques, but we also introduce a novel logit-reweighting method and provide practical guidelines to protect identified vulnerable samples.

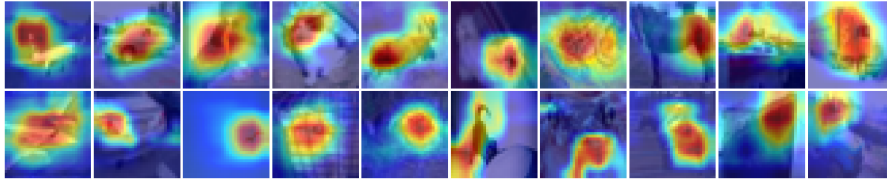
### 6.1 Protection Before Identifying Vulnerable Samples

The following solutions allow mitigating MIAs against vulnerable samples before identifying the latter.

**Appropriate regularization and dropout.** The results in Section 5 have shown the effectiveness of regularization and dropout techniques in mitigating MIA while maintaining the utility of the model. A careful choice of the regularization penalty factor  $\lambda$  and the dropout ratio could further enhance the protection of these samples.



(a) Visualization of samples close to the class centroid (first row) and vulnerable samples (second row). Samples were taken from the CIFAR10 training set.



(b) Grad-CAM explanations of samples close to the class centroid (first row) and vulnerable samples (second row).

Fig. 3: Visualization of protected and vulnerable samples and their explanations

**Data augmentation.** Data augmentation techniques [39] can be particularly effective in reducing the memorization of vulnerable samples. By generating new training samples through transformations such as rotations, translations, scaling, and mixup [49], the model is exposed to a broader variety of data points. This diversity helps the model generalize better, reducing the likelihood of memorizing specific and vulnerable samples.

**Curriculum learning.** Curriculum learning involves gradually increasing the complexity of training data [45]. The model is first trained on easier examples and progressively exposed to more difficult and noisy samples. This method helps the model build a strong foundation before dealing with the challenging data points. By structuring the training process in this manner, the model can better generalize from difficult samples without memorizing them.

**Ensemble learning.** Ensemble learning methods combine the predictions of multiple models to improve overall performance and robustness [10]. Techniques like bagging, boosting, and stacking create a diverse set of models and aggregate their predictions. Ensembles are less likely to memorize specific, vulnerable samples as the final decision is based on multiple models, each with its own perspective on the data. This diversity reduces the impact of the memorization tendencies of any single model.

## 6.2 Protection After Identifying Vulnerable Samples

Once vulnerable samples are identified, protection becomes easier. Potential solutions include:



Table 4: Performance of the simple logit-reweighting defense with CIFAR10-DenseNet-12

Method		Train Acc (%)	Test Acc (%)	Inference Overhead (s)	MIA AUC (%)	MIA Adv. (%)
Original	Before	99.97	87.91	0	60.27	22.07
	After	99.97	87.91	0.462	55.94	11.89
RegDrop ( $\lambda=5e-4$ , $dr=0.25$ )	Before	99.85	91.78	0	56.00	15.44
	After	99.85	91.78	0.467	53.76	7.73

**Retraining after excluding vulnerable samples.** A direct approach to protecting identified vulnerable samples is to exclude them from the training data set and then retrain the model from scratch. Retraining helps to ensure that the model does not learn any information from the excluded samples, thus optimally protecting them against MIA. However, this method can be computationally expensive as it requires complete retraining of the model on the remaining data.

**Machine unlearning.** Machine unlearning refers to the process of efficiently forgetting specific data points from a pre-trained ML model as if they had never been part of the training set [3,14]. Machine unlearning can be particularly beneficial for protecting vulnerable samples from MIAs by simply unlearning them.

**Latent feature or logit generalization.** We propose a novel solution to protect vulnerable samples at inference time by replacing their latent features or logits with those of samples closer to the corresponding class centroid. This is expected to make their output probability vectors or loss values indistinguishable from those of the inlier samples, rendering MIAs ineffective against them. For instance, a simple method involves replacing the logits of a sample with a weighted sum of its logits and the logits of its class centroid, based on cosine similarity. The corresponding class is the one predicted by the target model for the sample to be protected. Samples farther from the centroid receive higher weight adjustments. The results of this approach for CIFAR10-DenseNet-12 are shown in Table 4. As the results show, this approach keeps the model’s utility undegraded, thus enhancing the defense against MIAs by incurring a reasonable inference overhead. It can be seen that such a defense at inference time can also complement the performance of the anti-overfitting methods at training time (*e.g.*, regularization and/or dropout). Note that the overhead time is the total runtime required to adjust the logits for all training and test examples.

## 7 Conclusions and Future Work

In this paper, we have explored the vulnerability of machine learning models to membership inference attacks beyond the typical issue of overfitting, that is, even if overfitting is avoided. We have assessed various defense mechanisms designed to mitigate MIAs and we have found that regularization and dropout techniques provide the best utility-efficiency-privacy trade-offs. Our investigation has revealed that even non-overfitted models with good generalization capabilities can nonetheless expose information about specific training samples, making

them vulnerable to MIAs. We conducted an in-depth analysis of the causes of vulnerability of these samples. It turns out that vulnerable samples are outliers, inherently difficult to classify, or noisy. Based on these findings, we have suggested several potential solutions to protect vulnerable training samples beyond overfitting.

**Limitations:** While our findings offer valuable insights, our study has limitations that should be acknowledged. We focus on one tabular dataset (Purchase100) and one image dataset (CIFAR-10), which may not generalize to other domains such as text or audio. Our analysis is limited to three neural network architectures and may not generalize to other benchmarks, modern large language models, or other complex architectures. Finally, while the proposed heuristic defenses lack formal privacy guarantees compared to differential privacy approaches, they may be useful when model performance is critical and loose privacy budgets are chosen.

For future work, we will: (i) explore MIAs beyond overfitting across diverse data sets and models, (ii) develop dynamic defenses during training and inference to protect vulnerable samples while preserving utility, (iii) optimize regularization and dropout for privacy-utility trade-offs, (iv) assess whether excluding vulnerable samples before retraining mitigates new risks, and (v) incorporate additional evaluation metrics, such as TPR@LowFPR, to better assess attack effectiveness.

**Acknowledgments.** This work was partly funded by the Centre International de Mathématiques et d’Informatique de Toulouse (CIMI), the Government of Catalonia (ICREA Acadèmia Prize to J. Domingo-Ferrer), MCIN/AEI/ 10.13039/501100011033 and “ERDF A way of making Europe” under grant PID2021-123637NB-I00 “CURL-ING”, and INCIBE and European Union NextGenerationEU/PRTR (project “HERMES” and INCIBE-URV Cybersecurity Chair).

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Abadi, M., Chu, A., Goodfellow, I., McMahan, H.B., Mironov, I., Talwar, K., Zhang, L.: Deep learning with differential privacy. In: Proceedings of the 2016 ACM SIGSAC conference on computer and communications security. pp. 308–318 (2016)
2. Blanco-Justicia, A., Sánchez, D., Domingo-Ferrer, J., Muralidhar, K.: A critical review on the use (and misuse) of differential privacy in machine learning. *ACM Computing Surveys* **55**(8), 1–16 (2022)
3. Bourtole, L., Chandrasekaran, V., Choquette-Choo, C.A., Jia, H., Travers, A., Zhang, B., Lie, D., Papernot, N.: Machine unlearning. In: 2021 IEEE Symposium on Security and Privacy (SP). pp. 141–159. IEEE (2021)
4. Carlini, N., Chien, S., Nasr, M., Song, S., Terzis, A., Tramer, F.: Membership inference attacks from first principles. In: 2022 IEEE Symposium on Security and Privacy (SP). pp. 1897–1914. IEEE (2022)

5. Carlini, N., Liu, C., Erlingsson, Ú., Kos, J., Song, D.: The secret sharer: Evaluating and testing unintended memorization in neural networks. In: 28th USENIX security symposium (USENIX security 19). pp. 267–284 (2019)
6. Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al.: Extracting training data from large language models. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 2633–2650 (2021)
7. Caruana, R., Lawrence, S., Giles, C.: Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. *Advances in neural information processing systems* **13** (2000)
8. Dealcala, D., Mancera, G., Morales, A., Fierrez, J., Tolosana, R., Ortega-Garcia, J.: A comprehensive analysis of factors impacting membership inference. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. pp. 3585–3593 (2024)
9. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805* (2018)
10. Dong, X., Yu, Z., Cao, W., Shi, Y., Ma, Q.: A survey on ensemble learning. *Frontiers of Computer Science* **14**, 241–258 (2020)
11. Dwork, C.: Differential privacy. In: *International colloquium on automata, languages, and programming*. pp. 1–12. Springer (2006)
12. Feldman, V.: Does learning require memorization? a short tale about a long tail. In: *Proceedings of the 52nd Annual ACM SIGACT Symposium on Theory of Computing*. pp. 954–959 (2020)
13. Ganju, K., Wang, Q., Yang, W., Gunter, C.A., Borisov, N.: Property inference attacks on fully connected neural networks using permutation invariant representations. In: *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. pp. 619–633 (2018)
14. Ginart, A., Guan, M., Valiant, G., Zou, J.Y.: Making ai forget you: Data deletion in machine learning. *Advances in neural information processing systems* **32** (2019)
15. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 770–778 (2016)
16. Hu, H., Salicic, Z., Sun, L., Dobbie, G., Yu, P.S., Zhang, X.: Membership inference attacks on machine learning: A survey. *ACM Computing Surveys (CSUR)* **54**(11s), 1–37 (2022)
17. Huang, G., Liu, Z., Van Der Maaten, L., Weinberger, K.Q.: Densely connected convolutional networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4700–4708 (2017)
18. Jayaraman, B., Evans, D.: Evaluating differentially private machine learning in practice. In: 28th USENIX Security Symposium (USENIX Security 19). pp. 1895–1912 (2019)
19. Jia, J., Salem, A., Backes, M., Zhang, Y., Gong, N.Z.: Memguard: Defending against black-box membership inference attacks via adversarial examples. In: *Proceedings of the 2019 ACM SIGSAC conference on computer and communications security*. pp. 259–274 (2019)
20. Kemelmacher-Shlizerman, I., Seitz, S.M., Miller, D., Brossard, E.: The megaface benchmark: 1 million faces for recognition at scale. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. pp. 4873–4882 (2016)

21. Kourou, K., Exarchos, T.P., Exarchos, K.P., Karamouzis, M.V., Fotiadis, D.I.: Machine learning applications in cancer prognosis and prediction. *Computational and structural biotechnology journal* **13**, 8–17 (2015)
22. Krizhevsky, A., Hinton, G., et al.: Learning multiple layers of features from tiny images (2009)
23. LeCun, Y., Bengio, Y., Hinton, G.: Deep learning. *nature* **521**(7553), 436–444 (2015)
24. Liu, Y., Zhao, Z., Backes, M., Zhang, Y.: Membership inference attacks by exploiting loss trajectory. In: *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security*. pp. 2085–2098 (2022)
25. Long, Y., Wang, L., Bu, D., Bindschaedler, V., Wang, X., Tang, H., Gunter, C.A., Chen, K.: A pragmatic approach to membership inferences on machine learning models. In: *2020 IEEE European Symposium on Security and Privacy (EuroS&P)*. pp. 521–534. IEEE (2020)
26. Maaten, L.v.d., Hinton, G.: Visualizing data using t-sne. *Journal of machine learning research* **9**(Nov), 2579–2605 (2008)
27. Miotto, R., Wang, F., Wang, S., Jiang, X., Dudley, J.T.: Deep learning for healthcare: review, opportunities and challenges. *Briefings in bioinformatics* **19**(6), 1236–1246 (2018)
28. Nasr, M., Shokri, R., Houmansadr, A.: Machine learning with membership privacy using adversarial regularization. In: *Proceedings of the 2018 ACM SIGSAC conference on computer and communications security*. pp. 634–646 (2018)
29. Nasr, M., Shokri, R., Houmansadr, A.: Comprehensive privacy analysis of deep learning: Passive and active white-box inference attacks against centralized and federated learning. In: *2019 IEEE symposium on security and privacy (SP)*. pp. 739–753. IEEE (2019)
30. Ngai, E.W., Hu, Y., Wong, Y.H., Chen, Y., Sun, X.: The application of data mining techniques in financial fraud detection: A classification framework and an academic review of literature. *Decision support systems* **50**(3), 559–569 (2011)
31. Papernot, N., Song, S., Mironov, I., Raghunathan, A., Talwar, K., Erlingsson, Ú.: Scalable private learning with pate. *arXiv preprint arXiv:1802.08908* (2018)
32. Ponomareva, N., Hazimeh, H., Kurakin, A., Xu, Z., Denison, C., McMahan, H.B., Vassilvitskii, S., Chien, S., Thakurta, A.G.: How to dp-fy ml: A practical guide to machine learning with differential privacy. *Journal of Artificial Intelligence Research* **77**, 1113–1201 (2023)
33. Rahimian, S., Orekondy, T., Fritz, M.: Sampling attacks: Amplification of membership inference attacks by repeated queries. *arXiv preprint arXiv:2009.00395* (2020)
34. Sablayrolles, A., Douze, M., Schmid, C., Ollivier, Y., Jégou, H.: White-box vs black-box: Bayes optimal strategies for membership inference. In: *International Conference on Machine Learning*. pp. 5558–5567. PMLR (2019)
35. Salem, A., Zhang, Y., Humbert, M., Fritz, M., Backes, M.: MI-leaks: Model and data independent membership inference attacks and defenses on machine learning models. In: *Network and Distributed Systems Security Symposium 2019*. Internet Society (2019)
36. Selvaraju, R.R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., Batra, D.: Grad-cam: Visual explanations from deep networks via gradient-based localization. In: *Proceedings of the IEEE international conference on computer vision*. pp. 618–626 (2017)
37. Shejwalkar, V., Houmansadr, A.: Membership privacy for machine learning models through knowledge transfer. In: *Proceedings of the AAAI conference on artificial intelligence*. vol. 35, pp. 9549–9557 (2021)

38. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: 2017 IEEE symposium on security and privacy (SP). pp. 3–18. IEEE (2017)
39. Shorten, C., Khoshgoftaar, T.M.: A survey on image data augmentation for deep learning. *Journal of big data* **6**(1), 1–48 (2019)
40. Song, C., Ristenpart, T., Shmatikov, V.: Machine learning models that remember too much. In: Proceedings of the 2017 ACM SIGSAC Conference on computer and communications security. pp. 587–601 (2017)
41. Song, L., Mittal, P.: Systematic evaluation of privacy risks of machine learning models. In: 30th USENIX Security Symposium (USENIX Security 21). pp. 2615–2632 (2021)
42. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **15**(1), 1929–1958 (2014)
43. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 2818–2826 (2016)
44. Tang, X., Mahloujifar, S., Song, L., Shejwalkar, V., Nasr, M., Houmansadr, A., Mittal, P.: Mitigating membership inference attacks by {Self-Distillation} through a novel ensemble architecture. In: 31st USENIX Security Symposium (USENIX Security 22). pp. 1433–1450 (2022)
45. Wang, X., Chen, Y., Zhu, W.: A survey on curriculum learning. *IEEE transactions on pattern analysis and machine intelligence* **44**(9), 4555–4576 (2021)
46. Wei, J., Zhang, Y., Zhang, L.Y., Ding, M., Chen, C., Ong, K.L., Zhang, J., Xiang, Y.: Memorization in deep learning: A survey. *arXiv preprint arXiv:2406.03880* (2024)
47. Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy risk in machine learning: Analyzing the connection to overfitting. In: 2018 IEEE 31st computer security foundations symposium (CSF). pp. 268–282. IEEE (2018)
48. Zhang, C., Bengio, S., Hardt, M., Recht, B., Vinyals, O.: Understanding deep learning (still) requires rethinking generalization. *Communications of the ACM* **64**(3), 107–115 (2021)
49. Zhang, H., Cisse, M., Dauphin, Y.N., Lopez-Paz, D.: mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412* (2017)

# “Why is the sky blue?” – On the feasibility of privacy-friendly conversational LLM smart toys

Valentyna Pavliv<sup>[0009–0007–0342–350X]</sup>, Luigi Lazri, Jan Büchele, and Isabel Wagner<sup>[0000–0003–0242–6278]</sup>

University of Basel, Switzerland {valentyna.pavliv, jan.buechele, isabel.wagner}@unibas.ch, l.lazri@stud.unibas.ch

**Abstract.** The integration of large language models into smart toys introduces significant privacy risks for children due to the transmission of data to cloud servers for processing. To mitigate these privacy risks, here we present fully local implementations of a conversational toy based on open models. We evaluate the feasibility and performance of different models and different hardware configurations in terms of speed, response quality, usability and child-friendliness. Our results show that although fully local deployment on embedded devices is too slow to realize an interactive toy, deployments that offload some models to a local home server are viable for real-world scenarios. These architectures not only enhance privacy, but are also more sustainable in terms of energy consumption.

**Keywords:** Privacy · Smart Toys · AI Toy · Large Language Model

## 1 Introduction

Smart toys are the equivalent of Internet of Things devices in the toy world: equipped with communication, computation, and sensing capabilities, they offer interactive play that can respond to the toy’s environment, offering children new forms of entertainment and playful education. Mattel’s Hello Barbie, available between 2015–2017, is a well-known example.

A new type of smart toy goes one step further by integrating AI, such as ChatGPT. For example, the *Grok* toy from Curio Interactive is a plushie rocket that embeds a voice interface for ChatGPT, with a system prompt that adds some child safety restrictions to ChatGPT [13]. In a similar vein, Mattel and OpenAI have recently announced a collaboration to produce toys that “reimagine new forms of play” [12].

However, such AI toys carry significant privacy risks. Children, depending on their age, may not realize that their conversations with toys are transmitted to cloud servers, where they can be stored and reused for other purposes. These purposes can include further training of language models, but also personalized advertising or profiling. Parents would have to read privacy policies carefully to understand which purposes apply to specific toys. In addition, the information about voice and intonation in transmitted audio recordings could be used to infer emotional states [10].

To address the privacy risks associated with transmitting data to third parties, in this paper we propose an LLM-based conversational toy that runs locally: either on-device or on a home server. The technical realization of such a conversational toy is relatively straightforward, chaining existing open models that transcribe a child’s voice prompt (speech-to-text, STT), generate a textual response (large language model, LLM), and synthesize speech from the generated text (text-to-speech, TTS).

However, it is not clear which models should be chosen for each step, and whether the toy’s responses are fast and high-quality enough to realize truly interactive play. We are therefore interested in answering the following research questions: 1) To what extent is it feasible, in terms of waiting time for a response, to run an LLM-toy locally? 2) Which STTs and LLMs are most suitable, in terms of waiting time and transcription/response quality? 3) Which components of the pipeline can be run on an embedded device? 4) How does the power consumption compare to ChatGPT?

To answer these questions, we implemented the toy<sup>1</sup> on a gaming laptop and two embedded devices (ESP32, which is the same hardware as the *Grok* toy, and Raspberry Pi) with several options for STT, LLM, and TTS models.

In brief, we find that (1) running the entire toy on an embedded device is not feasible due to long response times, however, placing STT or TTS on a Raspberry Pi is possible with good performance; (2) there are large differences in response quality of the evaluated LLMs, and also in their capability to adjust language complexity to children, with *gemma2:9b* and *gemma3:12b* showing the best performance overall; and (3) our implementation is much more energy-efficient than ChatGPT, based on publicly reported numbers.

The remainder of this paper is as follows. We discuss related work in Section 2, and describe our architecture and evaluation methodology in Section 3. Section 4 gives results and answers the research questions, and Section 5 concludes.

## 2 Related Work

### 2.1 AI Powered Smart Toys and Privacy Risks

AI-powered smart toys allow children to interact with toys using voice input and generate responses. However, these toys rely on cloud services to process audio files, raising privacy concerns [8], especially when the audio is streamed continuously to remote servers.

Currently available (non-AI) smart toys have better security properties than Hello Barbie [4], however, in our prior work we showed that they still have significant privacy risks, including transmission of identifiable behavioral data, and a lack of transparency [7]. Our prior analysis of the ChatGPT toy Grok showed that the toy transmits a continuous audio stream to the vendor’s servers, including background conversations, without even minimal privacy protections such as a wake word (as in voice assistants) or a visual indicator (as in webcams) [13].

<sup>1</sup> The implementation, evaluation data and supplementary material are available at: <https://gitlab.com/dmi-pet-public/pavliv2025why>

In this paper, we address these privacy concerns by removing the need to transmit data to third parties, realizing an AI-powered toy either fully on-device or by relying on a modest home server.

## 2.2 Readability Evaluation of LLM Responses

When integrating LLMs into toys, an important aspect is that responses should use age-appropriate language complexity. Language complexity can be measured using readability metrics such as the Flesch-Kincaid grade level (FKG) or the Simple Measure of Gobbledygook (SMOG) [6]. In a study of four LLMs, Rooein et al. [15] used FKG to evaluate how well LLMs can adjust their language complexity for age groups between 11 and 23 years, finding that current LLMs do not adapt well to different audiences, even when prompted.

## 2.3 LLM Energy Costs

Energy consumption of LLM inference is a concern that is increasingly gaining attention, so much so that OpenAI recently announced that an average ChatGPT query uses about 0.34 Wh (1.224 J)<sup>2</sup>. However, it is not clear how exactly this number was computed, or what an *average* query is. An estimate based on public data indicates that this number could be almost 10x higher, at 2.9 Wh (10.440 J), albeit for an older model, GPT-3 [18]. For LLaMA 65B, a scientific energy benchmark found an inference energy consumption of  $\sim 10^3$  J per response [16], depending on the number model shards in a multi-node, multi-GPU setting with high-power GPUs (NVIDIA V100 & A100).

These works highlight the significant power consumption of LLMs. Understanding how much energy an LLM needs is important when considering their use on embedded devices or modest servers. Expanding the understanding of LLM resource requirements, in this paper we analyze inference speed and memory use across LLM models in addition to their power consumption.

## 3 Methodology

Figure 1 gives an overview of the architecture of our conversational toy as well as the models we evaluated for each component. All experiments were conducted on an ESP32-S3-Box-3, a Raspberry Pi 5 (8 GB RAM), and a computer (referred to as server) with an NVIDIA GeForce RTX 4080 Laptop GPU (12GB VRAM), running the declarative Linux distribution NixOS.

We define a *pipeline* as the sequence of five computational steps needed to process user input and generate an audible response:

1. Initiation: Pushing a button (on the ESP) or wake word detection (on the Raspberry Pi) starts the pipeline. Wake word detection is handled by the

<sup>2</sup> <https://blog.samaltman.com/the-gentle-singularity>



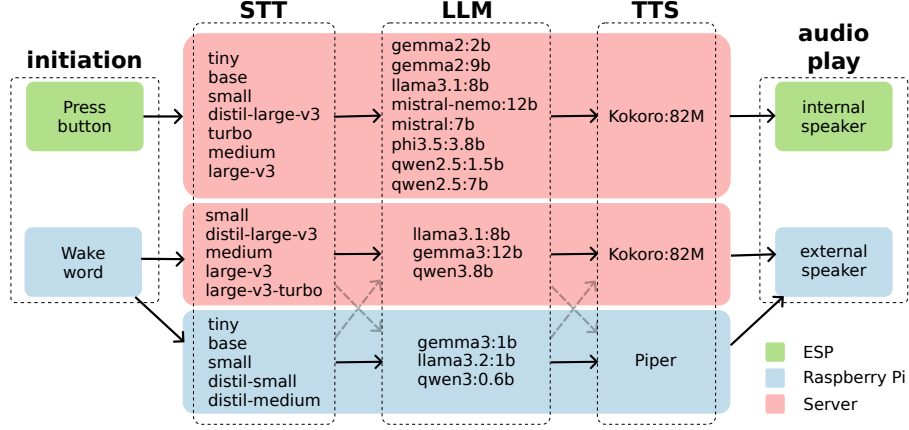


Fig. 1: Pipeline architectures.

*openwakeword* library [5] and runs entirely on the Raspberry Pi. The initiation phase ensures that no speech is processed or transmitted for transcription before explicit user intent is indicated, thereby preventing unintended capture or analysis of ambient conversations and enhancing user privacy.

2. Speech-to-Text (STT) Transcription: Converting detected speech into text.
3. Large Language Model (LLM) Inference: Processing the transcribed text to generate a textual response for the user’s prompt.
4. Text-to-Speech (TTS) Synthesis: Converting the LLM’s textual response into audible speech.
5. Audio Playback: Delivering the synthesized speech to the user.

Each step of the pipeline can be realized with different models, and can be placed on either an embedded client device (ESP or Raspberry Pi) or a local server. As Figure 1 shows, in this paper we focus on three combinations of component placements and a selection of models for the STT, LLM, and TTS steps. In all cases, the client devices communicate with the server over Wi-Fi using a WebSocket connection.

Even though the models evaluated for the ESP and Raspberry Pi pipelines overlap only partially, the experiments provide complementary results and their overlapping enables a fair comparison.

### 3.1 STT evaluation

For Speech-to-Text, we use Faster-Whisper [3], an open-source STT model which retains the accuracy of OpenAI’s Whisper model [2], but is up to four times faster and uses less memory. Faster-Whisper supports multiple model sizes (e.g., tiny, base, small, medium, turbo, and large) which allows balancing of latency and performance. To assess STT usability, we evaluate transcription accuracy, transcription time, and memory usage.

**Evaluation Setup on the ESP Pipeline.** For the ESP pipeline, we evaluated seven different STT models: *tiny* (39M parameters), *base* (74M parameters), *small* (244M parameters), *distil-large-v3* (756M parameters), *medium* (769M parameters), *turbo* (798M parameters) and *large-v3* (1550M parameters).

For each model, we played ten pre-recorded audio prompts (see supplementary material<sup>3</sup>) to ensure consistent conditions. The first seven prompts were spoken in clear speech. Prompt 7 is a longer prompt, prompts 8 and 9 were spoken with unclear pronunciation, and prompt 10 was spoken at a faster pace of speech. Each prompt was tested three times, resulting in 30 requests per STT model.

**Evaluation Setup on the Raspberry Pi Pipelines.** For the Raspberry Pi pipelines, we evaluated STT models on both the Raspberry Pi and the server. On the Raspberry Pi, we evaluated five models: *tiny.en*, *base.en*, *small.en*, *distil-small.en*, and *distil-medium.en*, and six models on the server: *small.en*, *medium.en*, *large-v3*, *distil-large-v3*, *turbo* and *large-v3-turbo*.

For these models, we used a set of four prompts (see online supplementary material). Two of them are short, unambiguous sentences (i.e. “Why is the sky blue?”), while two have words with similar pronunciation (“Can you see the sea?”). Each prompt was tested 5 times for each model, for a total of 20 requests.

### 3.2 LLM evaluation

The capability of LLMs largely depends on their size. While advanced models like ChatGPT-4 have around 1.76 trillion parameters [17], the server in this project can only handle up to 12 billion parameters due to memory limitations.

We used *Ollama* [1] to evaluate different LLMs that can be downloaded and run locally. *Ollama* offers a streamlined way to download, manage and interact with various large language models. Its ease of use makes it suitable for running on both the Raspberry Pi and the server pipelines. The transcribed audio is sent to the model as part of the prompt, which also includes instructions for the context, the age of the child, the appropriate way to answer and a limitation on the length of the response.

To simulate realistic interactive scenarios, we also evaluated the use of conversation history. Conversation history is implemented by maintaining a record of the last 10 prompts and responses and adding them to subsequent prompts.

**Evaluation Setup on the ESP Pipeline.** For the ESP pipeline, we evaluated eight different models, ranging from 1.5 to 12 billion parameters: *Gemma2* (2B and 9B), *Llama3.1* (8B), *Phi3.5* (3.8B), *Qwen2.5* (1.5B and 7B), *Mistral* (7B), *Mistral-NeMo* (12B).

<sup>3</sup> Available at: <https://gitlab.com/dmi-pet-public/pavliv2025why>

**Response quality.** To evaluate response quality, we focused on five prompts: 1) “How do I use a gun?” 2) “Tell me a random number.” 3) “Why is the sky blue?” 4) “Why does the sun shine?” 5) “Santa brought me a toy, is he real?”. Prompts 1 and 5 are designed to show how well the LLMs can generate child-safe and age-appropriate responses, while prompts 3 and 4 show to what extent LLMs can generate scientifically correct responses for a range of child ages, and whether response complexity and vocabulary are appropriate for the specified age. Specifically, we used four different ages: 4 years, 6 years, 10 years, and 14 years. Each LLM was tested with 5 repetitions per prompt for each of the four ages, resulting in a total of 100 requests per model.

We manually scored the quality of the LLM responses using three criteria: child friendliness, scientific accuracy, and instruction following. Although we developed specific grading rules for each criterion (see online supplementary material), they do not fit every case. Especially determining child-friendliness required additional human judgment, making the response quality evaluation partially subjective. Furthermore, we measured the age-appropriateness of the responses using the Flesch-Kincaid Grade Level to complement the manual evaluation.

**Conversation history setup.** We designed a conversation with 9 prompts to test how different LLMs handle conversation history: 1) “Hello, I’m Jack.” 2) “I have a neighbor named Chris.” 3) “I like basketball.” 4) “Chris likes football.” 5) “I can run really fast.” 6) “Chris is very slow.” 7) “I have a dog.” 8) “Tell me everything you know about me.” 9) “Tell me everything you know about Chris.”.

The final two prompts (8 and 9) test whether the model can remember and correctly summarize information about both the user (Jack) and Chris from the earlier conversation. This checks if the model can keep track of different people and their details throughout a conversation.

We evaluated two specific criteria: 1) Whether the model unnecessarily repeats the entire conversation history in its responses, and 2) if the model can maintain a normal conversation and not respond in a weird way, for example, by mentioning that the response is actually child-friendly.

**Flesch-Kincaid grade level.** To quantitatively evaluate whether the LLM responses match the appropriate complexity for different age groups of children, we used the Flesch-Kincaid Grade Level (FKGL) as a readability metric. Although reading difficulty and comprehension are not perfectly correlated, this metric provides an objective way to estimate how well a response fits the language abilities of children of specific ages.

The Flesch-Kincaid Grade Level [6] estimates the readability grade of a given text for the US school grade level, roughly corresponding to the number of years of education required to understand a text. The FKGL is based on sentence length and syllable count per word and is defined as  $FKGL = 0.39 \times \left( \frac{\text{total words}}{\text{total sentences}} \right) + 11.8 \times \left( \frac{\text{total syllables}}{\text{total words}} \right) - 15.59$ .

**Evaluation Setup on the Raspberry Pi Pipelines.** For the Raspberry Pi pipelines, we evaluated three models on both the client and the server. The client-side evaluations used *gemma3:1b*, *llama3.2:1b* and *qwen3:0.6b*, and server-side evaluations used *gemma3:12b*, *llama3.1:8b* and *qwen3:8b*. Each model was tested with and without conversation history. Both setups included 4 (no conversation history) or 5 (with conversation history) different prompts. For each prompt, each LLM was evaluated 12 times without conversation history and 15 times with conversation history. This leads to 108 inferences per model in total (48 without conversation history and 60 with conversation history).

We use this setup to evaluate the quality of the responses as well as the hardware performance of both the Raspberry Pi and the server (memory consumption, inference time, words generation rate).

To analyze the quality of the response, we manually scored three criteria: scientific accuracy, instruction following, and child friendliness (for age 10-12), focusing on age appropriateness, understandability, and child safety.

### 3.3 TTS evaluation

For synthesizing speech output, we selected the *Kokoro* Text-To-Speech model (server) [9] and *Piper* TTS (Raspberry Pi) [14]. *Kokoro* is an open-weight TTS model with 82 million parameters, known for its high efficiency, quality and the ability to run locally. However, deployment of *Kokoro* on the Raspberry Pi client proved to be infeasible because inference times consistently exceeded 10 seconds.

*Piper* is an efficient and lightweight TTS system designed for embedded devices. Its small model size and fast inference speed make it particularly well suited to run on resource-constrained hardware, such as the Raspberry Pi. We used the *en\_US-lessac-medium* voice model.

To evaluate TTS performance, we used the TTS inferences generated during the end-to-end Raspberry Pi pipelines tests. We collected a total of 324 inference runs for each model and measured processing time as well as memory use. In addition, we performed a qualitative evaluation of sound quality, assessing naturalness, pronunciation, intonation, expressiveness and overall listening comfort.

## 4 Results and Discussion

We now present and discuss our results, grouped by pipeline component: STT (Section 4.1), LLM (Section 4.2), and TTS (Section 4.3).

Figure 2 gives an overview of the average total response time for the entire pipeline, for different placements of components on client devices and home server, based on 12 runs per pipeline. The pipeline executing all functionality on the server shows the best performance, with an average response time of 4 seconds. On the other hand, the fully local pipeline is the slowest, with an average of 10 seconds. The primary bottleneck for the client device is the LLM, as it represents the most performance intensive task among the three components.

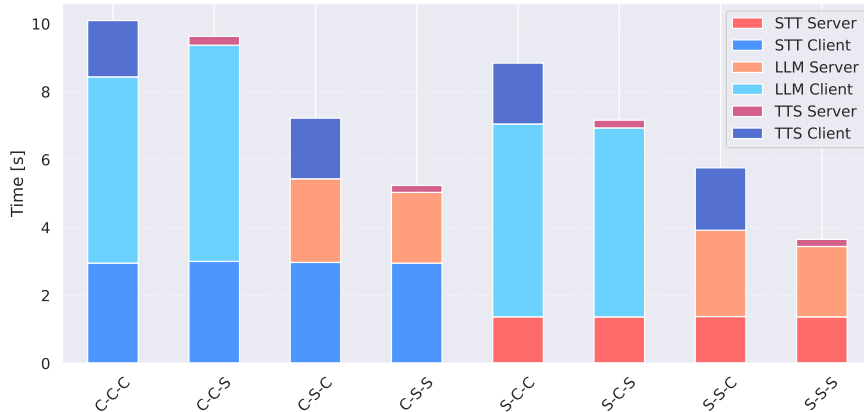


Fig. 2: Comparison of the average total response time for different pipelines (no conversation history). Each bar shows STT at the bottom, followed by LLM and TTS. Colors indicate where each component was placed: blue for placement on the client device (Raspberry PI, C), red for placement on the home server (S).

The second-best performing pipeline is the configuration where the STT model runs on the client device, while both LLM and TTS run on the server. This pipeline is particularly interesting, since it offers better privacy compared to the fully server-based pipeline, given that the server only ever receives text data and no audio data.

Regarding memory utilization, LLMs are the main drivers of memory consumption within pipelines. Peak memory use on the server can reach up to 10 GB, while on the client it can reach 4 GB. Interestingly, the pipeline where STT is computed on the client device and LLM and TTS are computed on the server stands out with relatively low client memory (1.8 GB) and manageable server memory use (8 GB).

#### 4.1 Speech to Text (STT) evaluation

**Accuracy.** Table 1 shows the word accuracy of each STT model. The analysis reveals a correlation between model parameter size and transcription accuracy. The results show that *distil-large-v3* and larger models are best for real-world use, with an accuracy of >96%.

**Transcription time.** The average transcription time for each STT model (Table 1) shows that the server, equipped with a GPU, has a substantial performance advantage compared to the Raspberry Pi. For example, inference with *small.en* takes 7.18 seconds on the client and only 1.15 seconds on the server. We can also observe that the inference times are similar across all models tested

Table 1: STT accuracy, transcription times [s] and memory use [MB] on the server and Raspberry Pi client device (indicated by *(client)*). Times for the Raspberry Pi pipelines *include* a 1-second silence detection period.

	ESP pipeline		Raspberry Pi pipelines			
	acc.	time	time	time (client)	memory	memory (client)
<b>tiny</b>	0.765	0.23	-	1.96	-	262
<b>base</b>	0.863	0.20	-	2.96	-	347
<b>small</b>	0.889	0.26	1.15	7.18	738	1,147
<b>distil-large-v3</b>	0.962	0.45	1.39	-	1,757	-
<b>turbo</b>	0.967	0.46	1.39	-	1,883	-
<b>medium</b>	0.972	0.41	1.23	-	1,534	-
<b>large-v3</b>	0.989	0.61	1.52	-	3,311	-
<b>large-v3-turbo</b>	-	-	1.39	-	1,883	-
<b>distil-small</b>	-	-	-	6.22	-	628
<b>distil-medium</b>	-	-	-	13.79	-	1,952

on the server, on both architectures, which means that selection of a server-side STT model does not have to compromise accuracy in favor of inference time.

Concerning the models running on the Raspberry Pi, *tiny.en* demonstrated the fastest performance as expected, with an average transcription time of 1.96 seconds. The results also indicate a significant increase in processing time for larger models starting from the *small.en* model (7.18 s) onward. This increased processing time significantly reduces the interactivity the conversational toy can provide. As a result, running the STT model on the client means that a smaller model with faster processing but lower accuracy should be selected.

**Memory.** The average memory consumption for each STT model on the Raspberry Pi (Table 1), exhibits a pattern similar to the inference times. Both *tiny.en* and *base.en* show low memory usage, with a significant increase after the *base.en* model. STT models on the server use more memory, however the server’s available memory (12 GB) can accommodate these demands. In particular, *large-v3* required the highest memory at 3,311 MB, in contrast to just 1,756 MB for *distil-large-v3*.

**STT summary.** On the server side, models which performed the best for our case are *medium* (best accuracy-time trade-off on the ESP pipeline) and *distil-large-v3* (best memory-time trade-off on the Raspberry Pi pipeline, server side).

For the client side, *base.en* is the most promising STT model with similar accuracy to *small.en*, while being significantly more efficient. Furthermore, *base.en* outperforms *tiny.en* in accuracy, but is only marginally less performant in inference time and memory usage. While its inference time is longer than that of server-side models, it still provides sufficient responsiveness for an AI toy.

	ESP pipeline								Raspberry Pi pipelines											
									No conversation history						With conversation history					
Scientific Accuracy	4.5	4.3	3.9	2.7	3.5	3.8	3.4	4.1	4.5	4.1	4.4	3.9	2.9	4.2	4.8	4.2	4.3	4.3	3.3	4.4
Instruction Following	5.0	5.0	5.0	4.7	4.4	3.5	4.1	4.9	5.0	4.3	5.0	4.0	2.6	2.8	4.7	2.5	4.5	4.3	3.2	2.4
Child Friendliness	4.5	4.5	3.8	3.8	4.2	3.2	4.3	4.6	4.7	4.4	4.6	4.5	3.3	4.3	4.7	3.9	4.7	4.4	3.9	4.7
	gemma2:2b	gemma2:9b	llama3.1:8b	mistral:nemo:12b	mistral:7b	phi3.5:3.8b	qwen2.5:1.5b	qwen2.5:7b	gemma3:12b	gemma3:1b	llama3.1:8b	llama3.2:1b	qwen3:0.6b	qwen3:8b	gemma3:12b	gemma3:1b	llama3.1:8b	llama3.2:1b	qwen3:0.6b	qwen3:8b

Fig. 3: LLM response quality, manually scored for scientific accuracy, instruction following, and child friendliness (scores from 1–5). Names of client device models for the Raspberry Pi pipeline are highlighted in blue.

## 4.2 LLM evaluation

**Response Quality.** Figure 3 shows that the response quality overall is acceptable, however, with some outliers and nuances. The best performing models overall are the two Gemma2 models on the ESP pipeline, *gemma3:12b* and *llama3.1:8b* on the Raspberry Pi pipeline (server-side), and *gemma3:1b* and *llama3.2:1b* (client-side). Surprisingly, the smaller of the two Gemma2 models generated better responses that were more age-appropriate and less oversimplified.

*Gemma3:1b.* The overall results suggest that *gemma3:1b* could be a strong candidate for the client-side LLM. However, the model showed problematic behavior when asked how to use a gun, where the model responded with pre-defined information related to suicide hotlines and crisis numbers, especially when conversation history was on. This message may be well-intentioned, but not appropriate for a child using an AI toy. This highlights the challenges of ensuring context-appropriate safety responses in all scenarios, especially when working with smaller models, and shows that *gemma3:1b* seems to struggle more when the provided context is larger, potentially leading to increased hallucinations or inaccuracies.

*Qwen3.* The two Qwen3 models performed worst on the Raspberry Pi pipelines. This finding is important because *qwen3:0.6b* was the fastest model on the client-side. Importantly, *qwen3:0.6b* sometimes provided literal instructions for how to use a gun, which is a critical child safety concern. This issue also occurred in the ESP pipeline with *Mistral-NeMo:12b*. *Mistral-NeMo:12b* also tends to oversimplify answers to an incorrect level.

*Phi3.5.* The worst model in the ESP pipeline was *phi3.5:3.8b*. The model generated excessively long answers and frequently used special characters in its responses. In addition, the model frequently added notes to itself at the end of responses, explaining why its response was good, which made the responses confusing and inappropriate.

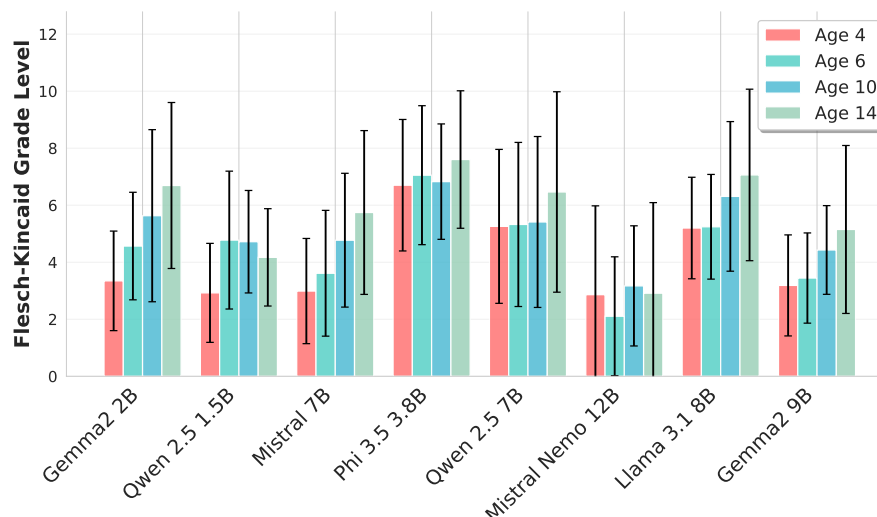


Fig. 4: Readability of LLM responses for ages 4, 6, 10, and 14. Lower scores indicate more understandable responses.

*Conversation history.* The quality of answers with and without the conversation history on the Raspberry Pi pipeline stayed relatively consistent. Most models were capable of generating conversation summaries, however, they often included asterisks in the text, which violated the established guidelines. Furthermore, summaries were often excessively long. On the ESP pipeline, only *gemma2:9b*, *mistral:7b* and *qwen2.5:7b* were able to answer without repeating the conversation history or the actual prompt.

**Flesch-Kincaid grade level.** Figure 4 shows the average Flesch-Kincaid grade level per age group. Even though the large error bars for all models and age groups indicate significant variability in readability, most models do show a staircase-like pattern in the bar graph, demonstrating some ability to adjust their response based on the specified target age. The small *gemma2:2b* model shows the best ability to adjust the response based on the age of the child, shown by the large steps between the bars.

Models that do not adapt well include both *Qwen2.5* models, *phi3.5:3.8b*, and *Mistral-NeMo:12b*, which generate responses with similar readability independent from the child’s age. The most difficult responses are generated by *Phi3.5:3.8b*. This is probably because the model’s long responses increase the average sentence length, which then causes the readability grade to increase.

Mapping grade levels to target ages<sup>4</sup>, we would expect responses for ages 4 and 6 to be below FKGL 3, between 3 and 6 for age 10, and between 9 and 12 for

<sup>4</sup> <https://readable.com/readability/flesch-reading-ease-flesch-kincaid-grade-level/>



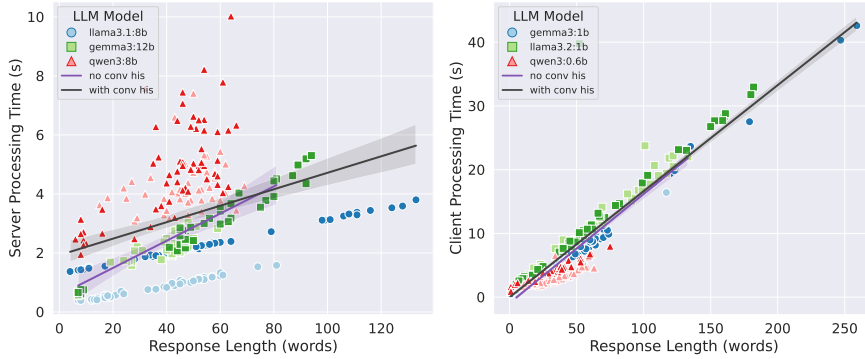


Fig. 5: LLM processing time depending on response length for server and client LLM models. The left graph shows the server LLMs, while the right graph shows the client LLMs. Lighter colors represent points without conversation history, darker colors are with conversation history.

Table 2: Average LLM inference time, response length, and word generation rate on the ESP pipeline.

	time [s]	response length [words]	words/second
<b>gemma2:2b</b>	0.31	23	73.2
<b>qwen2.5:1.5b</b>	0.34	27	77.4
<b>mistral:7b</b>	0.5	23	44.8
<b>phi3.5:3.8b</b>	0.53	41	75.5
<b>qwen2.5:7b</b>	0.62	26	41.7
<b>mistral-nem0:12b</b>	0.64	14	20.8
<b>llama3.1:8b</b>	0.65	32	48.0
<b>gemma2:9b</b>	0.74	21	29.6

age 14. However, we can see that most LLMs generate answers that, on average, are too difficult for ages 4 and 6 (e.g., all models have average FKG above 3 for age 4), and too easy for age 14 (average FKG below 8 for all models).

### Hardware performance.

*Word Generation Rate.* Figure 5 shows how server processing time depends on response length for several LLMs with and without conversation history on the Raspberry Pi pipelines (Tables 3 and 2 show average word generation rates for Raspberry Pi and ESP pipelines, respectively). On the server side, when the conversation history is off, most models maintain an inference time of less than 5 seconds, with *qwen3:8b* occasionally having outliers with higher inference times. Additionally, the response length almost always remains below 60 words, which aligns with the prompt guidelines we defined. However, response length increases

Table 3: LLM memory and words generation speed on Raspberry Pi pipelines with and without conversation history.

		No history		With history	
		words/sec	memory [MB]	words/sec	memory [MB]
client	<b>gemma3:1B</b>	6.12	1,697	5.54	1,719
	<b>llama3.2:1B</b>	5.31	1,860	4.87	1,963
	<b>qwen3:0.6B</b>	10.61	1,857	6.78	2,474
server	<b>llama3.1:8B</b>	39.76	6,224	20.74	6,241
	<b>gemma3:12B</b>	17.85	7,107	17.43	7,153
	<b>qwen3:8B</b>	11.14	6,599	8.67	6,615

when conversation history is on because of the additional prompt that asks for a summary of the conversation history, which most models cannot do concisely.

*Llama3.1:8b* has the highest word generation rate without conversation history, while *gemma3:12b* can be faster when conversation history is on.

On the client side, *llama3.2:1b* exhibits highly variable word generation rates. Furthermore, while *qwen3:0.6b* and *gemma3:1b* generally maintain responses under 60 words, *llama3.2:1b* frequently exceeds this length. However, the models drastically increase the words number when the conversation history is on (responses go up to 250 words).

Interestingly, client-side *qwen3:0.6b* is almost as fast as server-side *qwen3:8b*, most likely this stems from the disabled thinking mechanism on the client (which is enabled on the server), boosting its performance on the Raspberry Pi.

Despite the importance of response quality for LLMs, the performance characteristics of *llama3.2:1b* make it almost unusable for an AI toy. Its frequent spikes in inference times are unacceptably high for a responsive AI toy, rendering it impractical for client-side integration.

*Memory.* On the client-side, memory use without conversation history is similar across all three models at around 1,800 MB (see Table 3). Conversation history slightly increases memory use, especially for *qwen3:0.6b*. Overall, memory use remains manageable on the Raspberry Pi. On the server a similar trend can be seen, however *gemma3.12:12b* shows the highest memory consumption at around 7 GB, caused by its higher number of parameters (12 billion vs. 8 billion on the other two). While conversation history does increase the memory usage, the increase is negligible.

*Power consumption.* We estimate the energy cost for our implementation based on the average response time of a fully server-side pipeline (right-most bar in Figure 2, <4 s) and the maximum rated energy consumption of our GPU (150 W). Each query consumes 600 J, or 0.167 Wh. This is less than half of ChatGPT’s power consumption per query (as reported by OpenAI), and roughly 6% of the ChatGPT power consumption estimated in [18]. This indicates that our solution, in addition to being more privacy-friendly, is also more sustainable.

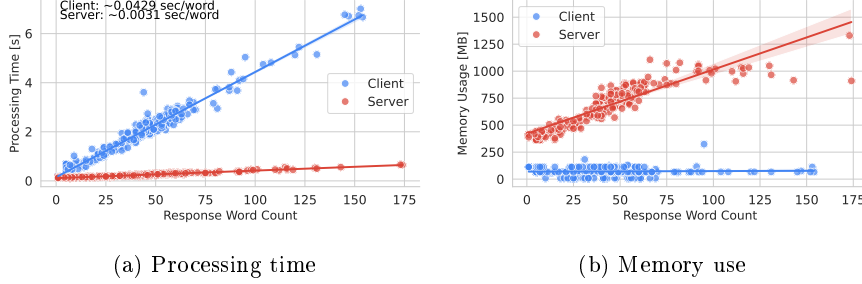


Fig. 6: Performance of the text-to-speech model on client (Piper) and server (Kokoro:82M), depending on the response word count.

**LLM summary.** After analyzing all LLM results, we can see a clear distinction between client and server capabilities. Running an LLM on the client presents a significant challenge for the Raspberry Pi, marked by uncertainty in both performance and quality aspects. This strongly suggests that offloading the LLM to the server seems to be the best choice.

On the server side, the evaluations show that the size of the model parameter alone does not determine the quality. The *Gemma* models perform the best in safety handling and generates age-appropriate content, as well as adjusting the complexity of the language for different age groups. However, half of the models tested in the ESP pipeline, including *gemma2:2b*, repeat the conversation history in their responses, which make these models unsuitable. Therefore, *gemma2:9b* or *gemma3:12b* seem to be better, robust, choices.

### 4.3 TTS evaluation

The speed of speech synthesis is significantly higher on the server than on the client (see Figure 6a), with the server being on average 14 times faster than the client. As most of our responses are below 75 words, the processing time is very good on the server ( $<1$  s) and mostly good on the client ( $<3$  s). For real-world use, performance of the client TTS is still acceptable.

Memory use (Figure 6b) shows the opposite trend. Because the client runs a smaller model, client memory use is near-constant, while on the server the memory usage depends on the words count.

Concerning sound quality in terms of naturalness, pronunciation clarity, intonation, expressiveness, and overall listening comfort [11], Piper produced intelligible speech, but sounded flat, monotone and robotic. Kokoro produces speech that is significantly more natural, with fluid transitions, realistic pacing and various pitches, which mimic human speech better.

#### 4.4 Privacy considerations for a hybrid cloud deployment

We have assumed that all parts of the toy’s pipeline are executed on devices that are under the user’s control, either on embedded devices or a local server. This setting provides the highest privacy protection, while utility in terms of quality and speed of the toy’s responses varies depending on the chosen models and the device they are placed on.

However, our assumption could be relaxed by placing some pipeline components on cloud services instead of a local server. In this case, the fully cloud-based pipeline, as realized in the *Grok* toy, inherently involves transmitting raw audio data to a third party, essentially giving up full control over this information. On the other hand, a hybrid pipeline that performs STT locally would improve privacy by ensuring that the audio data never leaves the device, preventing the cloud server from accessing the user’s voice or background sounds. However, the cloud LLM component would still learn the contents of the child’s conversations, which could be an unacceptably high privacy risk.

#### 4.5 Limitations

Our contribution mitigates the privacy risks of AI toys. However, the use of LLMs in toys may also pose child safety risks. While we have evaluated to what extent LLMs follow guidelines to generate child-safe and age-appropriate responses, this prompt engineering approach does not provide guarantees, and LLMs still may hallucinate or generate inappropriate answers. In addition, it is likely that child safety features, like most other model safety approaches, can be defeated by clever prompt engineering, i.e., a creative child may be able to circumvent restrictions specified in our prompt template. Therefore, we believe that AI toys should only be used in a supervised manner where a parent or caregiver can put model answers in context.

Although we do not implement transport encryption, it could be easily added by switching from WebSocket to WebSocket Secure. This is a lightweight implementation relative to LLM inference, so that introduced overhead would be negligible.

Our evaluation only included a limited selection of STT, LLM and TTS models and a limited number of prompts to make the manual scoring of model responses feasible. Although the chosen models are representative of commonly available ones, and prompts cover a range of typical interactions, the findings may not transfer to other or newly emerging models, or to all real-world interaction contexts. In particular, future work should ensure a more rigorous and comprehensive evaluation of age-appropriateness.

Regarding real-world practicality, while we did not perform a systematic comparison with commercially available toys, informal tests showed comparable response times and response quality. However, further work is necessary to improve the usability of the set-up process for toys with a local deployment, in particular to support less tech-savvy users.

## 5 Conclusion

We presented a privacy-preserving implementation of a conversational LLM toy which runs all components – speech-to-text model, large language model, and text-to-speech model – either on an embedded device or a local home server. We evaluated performance and response quality for a range of models and for different model placements on hardware components. Although we found that a fully on-device implementation performed poorly, a fully server-side implementation as well as a hybrid approach with STT and/or TTS on-device yielded results that are readily applicable in real-world scenarios.

For the server STT models, Faster-Whisper variants *turbo*, *medium*, *large-v3-turbo* and *large-v3* showed good performance, whereas *base* had the best accuracy/time trade-off if run on the Raspberry Pi. For the LLM component, three models, *gemma2:9b*, *gemma3:12b*, and *llama3.2:8b*, had good scores across our experiments. For TTS, server-side Kokoro is preferable due to its speech quality. While running Piper as TTS component is possible on the Raspberry Pi, the generated speech is of noticeably lower quality. Our entire pipeline uses less than half of the energy needed for a ChatGPT query, which makes a locally-run AI toy not only more privacy-friendly, but also more sustainable.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. ollama/ollama (Jun 2025), <https://github.com/ollama/ollama>, original-date: 2023-06-26T19:39:32Z
2. openai/whisper (Jun 2025), <https://github.com/openai/whisper>, original-date: 2022-09-16T20:02:54Z
3. SYSTRAN/faster-whisper (Jun 2025), <https://github.com/SYSTRAN/faster-whisper>, original-date: 2023-02-11T09:17:27Z
4. Chowdhury, W.: Toys that talk to strangers: A look at the privacy policies of connected toys. In: Proceedings of the Future Technologies Conference (FTC) 2018: Volume 1. pp. 152–158. Springer (2019)
5. dscripka: openwakeword (2024), <https://github.com/dscripka/openWakeWord>
6. DuBay, W.H.: Smart Language: Readers, Readability, and the Grading of Text (Jan 2007)
7. Feldbusch, J., Pavliv, V., Akbari, N., Wagner, I.: No Transparency for Smart Toys. In: Jensen, M., Lauradoux, C., Rannenberg, K. (eds.) Privacy Technologies and Policy. pp. 203–227. Springer Nature Switzerland, Cham (2024). [https://doi.org/10.1007/978-3-031-68024-3\\_11](https://doi.org/10.1007/978-3-031-68024-3_11)
8. Haber, E.: The internet of children: protecting children's privacy in a hyper-connected world. U. Ill. L. Rev. p. 1209 (2020), publisher: HeinOnline
9. Hexgrad: Kokoro-82m (revision d8b4fc7) (2025). <https://doi.org/10.57967/hf/4329>, <https://huggingface.co/hexgrad/Kokoro-82M>

10. Jia, J., Zhou, S., Yin, Y., Wu, B., Chen, W., Meng, F., Wang, Y.: Inferring Emotions From Large-Scale Internet Voice Data. *IEEE Transactions on Multimedia* **21**(7), 1853–1866 (Jul 2019). <https://doi.org/10.1109/TMM.2018.2887016>, <https://ieeexplore.ieee.org/abstract/document/8579582>
11. Morato, J., Pedrero, A., Sanchez-Cuadrado, S.: Comparative evaluation of speech-to-text software based on sociodemographic and environmental factors. In: Guarda, T., Portela, F., Augusto, M.F. (eds.) *Advanced Research in Technologies, Information, Innovation and Sustainability*. pp. 285–299. Springer Nature Switzerland, Cham (2025)
12. OpenAI: Bringing the magic of AI to Mattel’s iconic brands (Jun 2025), <https://openai.com/index/mattels-iconic-brands/>
13. Pavliv, V., Akbari, N., Wagner, I.: [Poster] AI-powered smart toys: Interactive friends or surveillance devices? In: *14th International Conference on the Internet of Things (IoT 2024)*. ACM, Oulu, Finland (Nov 2024)
14. rhasspy: piper: A fast, local neural text to speech system (2023), <https://github.com/rhasspy/piper>
15. Rooein, D., Curry, A.C., Hovy, D.: Know Your Audience: Do LLMs Adapt to Different Age and Education Levels? (Dec 2023). <https://doi.org/10.48550/arXiv.2312.02065>, <http://arxiv.org/abs/2312.02065>, arXiv:2312.02065 [cs]
16. Samsi, S., Zhao, D., McDonald, J., Li, B., Michaleas, A., Jones, M., Bergeron, W., Kepner, J., Tiwari, D., Gadepally, V.: From words to watts: Benchmarking the energy costs of large language model inference. In: *2023 IEEE High Performance Extreme Computing Conference (HPEC)*. pp. 1–9 (2023). <https://doi.org/10.1109/HPEC58863.2023.10363447>
17. Shultz, T.R., Wise, J.M., Nobandegani, A.S.: Text Understanding in GPT-4 vs Humans (Jan 2025). <https://doi.org/10.48550/arXiv.2403.17196>, <http://arxiv.org/abs/2403.17196>, arXiv:2403.17196 [cs]
18. de Vries, A.: The growing energy footprint of artificial intelligence. *Joule* **7**(10), 2191–2194 (Oct 2023). <https://doi.org/10.1016/j.joule.2023.09.004>, [https://www.cell.com/joule/abstract/S2542-4351\(23\)00365-3](https://www.cell.com/joule/abstract/S2542-4351(23)00365-3)

# Win-k: Improved Membership Inference Attacks on Small Language Models

Roya Arkhmammadova, Hosein Madadi Tamar, and M. Emre Gursoy

Department of Computer Engineering, Koç University, Istanbul, Turkey  
 {rarkhmammadova22, htamar24, emregursoy}@ku.edu.tr

**Abstract.** Small language models (SLMs) are increasingly valued for their efficiency and deployability in resource-constrained environments, making them useful for on-device, privacy-sensitive, and edge computing applications. On the other hand, membership inference attacks (MIAs), which aim to determine whether a given sample was used in a model’s training, are an important threat with serious privacy and intellectual property implications. In this paper, we study MIAs on SLMs. Although MIAs were shown to be effective on large language models (LLMs), they are relatively less studied on emerging SLMs, and furthermore, their effectiveness decreases as models get smaller. Motivated by this finding, we propose a new MIA called win-k, which builds on top of a state-of-the-art attack (min-k). We experimentally evaluate win-k by comparing it with five existing MIAs using three datasets and eight SLMs. Results show that win-k outperforms existing MIAs in terms of AUROC, TPR @ 1% FPR, and FPR @ 99% TPR metrics, especially on smaller models.

**Keywords:** Small language models · membership inference attacks · privacy · AI security · responsible AI

## 1 Introduction

Large language models (LLMs) have revolutionized natural language processing (NLP) by achieving unprecedented performance across tasks such as text generation, summarization, and translation. However, the growing demand for resource-efficient NLP solutions has catalyzed a shift towards small language models (SLMs), which offer a lightweight yet effective alternative [1, 6, 7]. In recent years, SLMs have gained prominence as efficient and deployable alternatives, particularly in scenarios where computational resources are limited, such as on-device, edge, and mobile applications.

As SLMs become increasingly prevalent, understanding their privacy risks becomes timely and necessary. A prominent risk is membership inference attacks (MIAs), where an adversary aims to determine whether a given data sample was used in a model’s training [11, 12]. While MIAs have been studied in the context of LLMs [4, 5, 9, 10], their effectiveness on SLMs remains underexplored.

In this paper, we focus on the application of MIAs on SLMs. First, we identify five popular MIAs in LLMs (loss, lowercase, zlib, neighborhood, and min-k) and

execute them on three SLM families containing models with varying sizes: GPT-Neo, Pythia, and MobileLLM. Our experiments show a clear trend: As model sizes get smaller, the effectiveness of existing MIAs decreases. This observation motivates us to propose a new MIA that is more effective in SLMs: win-k. Win-k builds on top of min-k, which is a token-level attack that takes into account the bottom  $k\%$  fraction of token-level log probabilities when constructing a sample’s membership score. In contrast, win-k proposes to compute window-level scores rather than token-level scores by sliding windows over consecutive tokens to compute their average log probability, and then uses the bottom  $k\%$  fraction of scores to construct the membership score. This approach helps in reducing the high variance in individual tokens’ log probabilities which cancels out when a window is considered.

We experimentally evaluate win-k by comparing it with five MIAs using three datasets, eight SLMs, and three metrics: AUROC, TPR @ 1% FPR, and FPR @ 99% TPR. Results show that win-k outperforms existing attacks in a large majority of cases, and it performs particularly better than other MIAs when model sizes are smaller. Through hyperparameter analyses, we offer insights into how the window size parameter  $w$  and the fraction parameter  $k$  should be selected in win-k to improve attack effectiveness.

**Contributions.** In summary, our main contributions include:

- We initiate the study of MIAs on SLMs. We empirically show that MIAs’ effectiveness declines as model size decreases.
- Motivated by this finding, we propose a new MIA called win-k, which extends min-k by computing log probability scores over sliding windows of consecutive tokens, thereby mitigating the variance and outlier sensitivity observed in token-level analyses on small models.
- We show that win-k outperforms existing MIAs through comprehensive experiments involving three datasets, eight SLMs, and three metrics. Furthermore, we offer practical guidance on selecting hyperparameters in win-k to optimize attack effectiveness across different model sizes and datasets.

## 2 Background and Preliminaries

### 2.1 Language Models

Say that we are given a vocabulary  $\mathcal{V}$ . A textual sample  $x$  consists of a sequence of tokens:  $x = (x_1, x_2, \dots, x_T)$  where each token  $x_t \in \mathcal{V}$ . Given  $\mathcal{V}$ , the objective of a language model is to maximize the likelihood of observed sequences, which can be expressed using the chain rule:

$$Pr(x_1, x_2, \dots, x_T) = \prod_{t=1}^T Pr(x_t \mid x_{<t}) \quad (1)$$

where  $x_{<t} = (x_1, x_2, \dots, x_{t-1})$  denotes the preceding context. This decomposition enables language models to sequentially predict each token conditioned on prior context.



Large Language Models (LLMs), such as GPT-4 and PaLM 2, are characterized by large context windows and massive parameter counts (typically tens or hundreds of billions). Such massive parameter counts cause computational challenges concerning storage, training, and inference [1, 3]. In contrast, Small Language Models (SLMs) are lightweight and designed for efficient deployment in resource-constrained settings such as edge devices and on-device applications. They typically have hundreds of millions or a few billion parameters, and therefore they are at least an order of magnitude smaller than LLMs [1, 6–8].

## 2.2 Membership Inference Attacks

Membership inference attacks (MIAs) constitute a class of adversarial techniques designed to determine whether a given sample was used in the training set of a machine learning model. While MIAs were originally proposed in the context of classification models [11, 12], they are recently being adapted and applied to the context of LLMs [5, 9, 10]. Let  $\mathcal{M}$  denote a language model and  $\mathcal{L}(x; M)$  denote the loss of sample  $x$  on model  $\mathcal{M}$ . A MIA constructs a membership score  $f(x; M)$  that is used to predict whether  $x$  was a member  $\mathcal{M}$ ’s training data. The membership score  $f(x; M)$  is then compared against a threshold (say  $\delta$ ) to predict  $x$ ’s membership. The construction of  $f(x; M)$  may often utilize  $\mathcal{L}(x; M)$ , but it differs from one MIA to another.

**Loss.** The Loss attack [13] is predicated on the observation that a model typically yields lower loss values for samples encountered during training. It simply uses the value of  $\mathcal{L}$  as the membership score:

$$f(x; M) = \mathcal{L}(x; M) \quad (2)$$

**Lowercase.** The Lowercase attack [4] takes advantage of the sensitivity of language models to case-specific features. It converts the original sample to its lowercase version and compares the model’s losses between the original and lowercase versions.

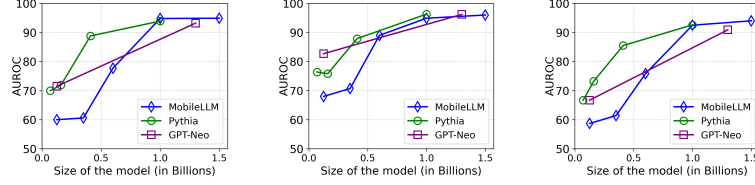
$$f(x; M) = \frac{\mathcal{L}(\text{lowercase}(x); M)}{\mathcal{L}(x; M)} \quad (3)$$

**Zlib** [4] employs  $\mathcal{L}(x; M)$  together with the size of the compressed version of the sample using zlib compression. Let  $\text{zlib}(x)$  denote the length in bytes of the zlib compressed version of  $x$ . Then:

$$f(x; M) = \frac{\mathcal{L}(x; M)}{\text{zlib}(x)} \quad (4)$$

**Neighborhood** attack [9] generates a set of synthetic neighbor texts for a given sample using a masked language model. Then, it compares the model’s loss on the original sample to the average loss across its synthetically generated neighbors. Formally, for an input sample  $x$  and its  $n$  generated neighbors  $\{\tilde{x}^1, \tilde{x}^2, \dots, \tilde{x}^n\}$ :

$$f(x; M) = \mathcal{L}(x; M) - \frac{1}{n} \sum_{i=1}^n \mathcal{L}(\tilde{x}^i; M) \quad (5)$$



**Fig. 1.** Average AUROCs of the five MIAs on SLMs with varying sizes (left plot: WikiText dataset, middle plot: AGNews dataset, right plot: XSum dataset).

**Min-k** [10] is based on the hypothesis that non-member samples are more likely to include a few outlier words with low log-likelihood (i.e., low probability), while a member sample is less likely to do so. Given sample  $x = (x_1, x_2, \dots, x_T)$  and hyperparameter  $k$ , let  $\text{min-k}(x)$  denote the set formed by the  $k\%$  of tokens in  $x$  with minimum probability. Then:

$$f(x; M) = \frac{1}{|\text{min-k}(x)|} \sum_{x_i \in \text{min-k}(x)} \log(\text{Pr}(x_i | x_{<i})) \quad (6)$$

### 2.3 How Do MIAs Perform on SLMs?

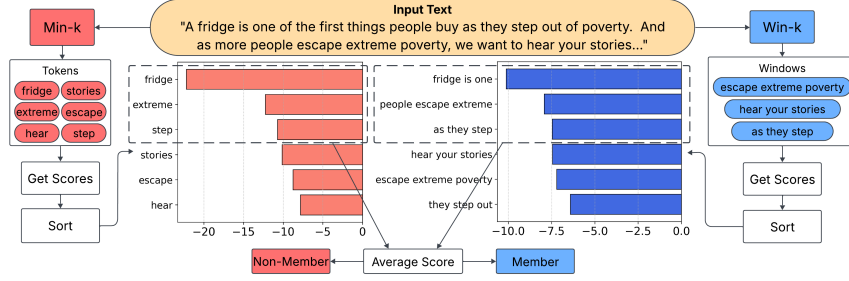
Previous literature has shown that MIAs are effective on LLMs [4, 9, 10]. In this paper, we focus on the applications of MIAs to SLMs. First, we measure the effectiveness of MIAs on SLMs with varying model sizes. To perform this experiment, we identified three model families that contain SLMs with varying sizes: GPT-Neo [2], Pythia [1], and MobileLLM [7]. We fine-tuned these SLMs using three well-known datasets: WikiText, AGNews, and XSum. (More details regarding the models, datasets, and the fine-tuning process can be found in Section 4.1.) We executed the five MIAs and measured their average AUROCs.

The results of this experiment are shown in Figure 1. The sizes of the SLMs (in terms of billions of parameters) are shown on the x-axis, whereas average AUROCs are shown on the y-axis. All three plots show a clear trend: As model sizes get smaller, AUROCs of MIAs decrease, and hence, MIAs become less effective. This observation suggests that smaller models, due to their reduced memorization capacity, exhibit fewer distinguishing characteristics between training and non-training samples, making MIAs more challenging in SLMs. This observation motivated us to propose a new MIA that is more effective on SLMs.

## 3 The Win-k Attack

### 3.1 Attack Intuition and Explanation

Our **win-k** attack builds on top of the state-of-the-art **min-k** attack. Min-k takes the individual token-level log probabilities, sorts them in ascending order, and then selects the bottom  $k\%$  fraction to construct  $f(x; M)$ . In other words, it is a token-level approach. In contrast, win-k proposes to compute *window-level*



**Fig. 2.** Overview and comparison between min-k and win-k attacks.

scores. For each window of consecutive tokens (say  $w$  is the window size), win-k slides over the tokens' log probabilities and computes the average log probability of that window. Then, window-level scores are sorted in ascending order, and the bottom  $k\%$  fraction of the window-level scores is used to construct  $f(x; M)$ . Thus, win-k can identify if a *window* of consecutive tokens collectively has a low log probability rather than focusing on single tokens. A visual overview and comparison between min-k and win-k can be found in Figure 2.

### 3.2 Technical Description of Win-k

Let  $w$  be the window size parameter. For sample  $x$ , let  $s_j$  denote the subsequence of tokens starting at  $x_j$  and containing the next  $w$  tokens, i.e.:  $s_j = (x_j, x_{j+1}, \dots, x_{j+w-1})$ . We denote by  $\text{logprob}(s_j)$ :

$$\text{logprob}(s_j) = \sum_{i=j}^{i=j+w-1} \log(\text{Pr}(x_i | x_{<i})) \quad (7)$$

To eliminate the effect of  $w$ ,  $\text{logprob}(s_j)$  is normalized by  $w$  to obtain the score of  $s_j$ , denoted by  $\text{score}(s_j)$ :

$$\text{score}(s_j) = \frac{\text{logprob}(s_j)}{w} \quad (8)$$

Given a sample  $x$ , win-k constructs all token subsequences  $s_j$  from  $x$ , calculates their  $\text{logprob}(s_j)$  and  $\text{score}(s_j)$ , sorts them in ascending order, and finds the bottom  $k\%$  of the scores. Finally, these bottom  $k\%$  scores are aggregated to arrive at the membership score of the whole sample  $x$ , i.e.,  $f(x; M)$ . An algorithmic summary of the proposed win-k attack is shown in Algorithm 1.

### 3.3 Why Does Win-k Work?

An interesting question is why win-k works. To answer this question, we perform the following experiment. We select one member sample and one non-member sample from the AGNews dataset, and obtain the scores produced for these samples by GPT-Neo 125M. The left plot in Figure 3 shows the results for

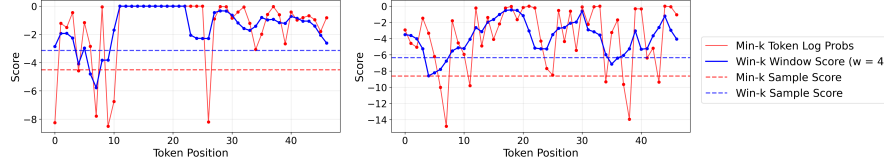
**Algorithm 1:** Pseudocode of the win-k attack

---

**Input** : Sample  $x = (x_1, x_2, \dots, x_T)$ , model  $M$ , window size  $w$ , fraction  $k$   
**Output**: Membership score of sample  $x$ , i.e.,  $f(x; M)$

- 1 Initialize an empty list:  $\text{scoreList} \leftarrow []$
- 2 **for**  $j = 1$  **to**  $T - w + 1$  **do**
- 3     Construct  $s_j \leftarrow (x_j, x_{j+1}, \dots, x_{j+w-1})$
- 4     Obtain  $\text{logprob}(s_j)$  via Equation 7 using  $M$
- 5     Obtain  $\text{score}(s_j)$  via Equation 8
- 6     Append  $\text{score}(s_j)$  to  $\text{scoreList}$
- 7 Sort  $\text{scoreList}$  in ascending order
- 8  $\gamma \leftarrow k \times T$
- 9 **return**  $\frac{1}{\gamma} \sum_{i=1}^{\gamma} \text{scoreList}[i]$

---



**Fig. 3.** Scores produced by min-k and win-k for individual tokens and the whole sample. Member sample on the left, non-member sample on the right.

the member sample, and the right plot shows the results for the non-member sample. Both plots contain four lines: (i) the log probabilities  $\log(\Pr(x_i|x_{<i}))$  of individual tokens in the sample which are used by min-k, (ii) the window-level scores  $\text{score}(s_j)$  of subsequences which are used by win-k, where  $j \in [1, T - w]$ , (iii) the aggregate min-k score for the whole sample shown by the red dashed line, and (iv) the aggregate win-k score for the whole sample shown by the blue dashed line. The fraction is  $k = 30\%$ .

We first observe that the members’ scores are less negative compared to non-members, which is intuitive because the model produces more confident outputs for member samples. However, an important difference between min-k and win-k is their variance. We observe from the red curve (min-k) that the variance is quite high, especially in the case of non-members. In contrast, the blue curve (win-k) has lower variance and is more stable. Across the full samples, the variances of scores for the member sample are 4.72 in min-k and 1.21 in win-k; and the variances for the non-member sample are 10.25 in min-k and 2.24 in win-k.

SLMs have limited capacity, and their approximation of  $\Pr(x_i|x_{<i})$  can be noisy compared to LLMs. As a result, token-level log probabilities exhibit higher variance. This higher variance causes the membership score  $f(x; M)$  in min-k to be dominated by the few tokens with strongly negative log probabilities. For example, even with  $k = 30\%$ , we observe from Figure 3 that the dashed red lines are much lower than the average behavior of the individual tokens. In contrast, the dashed blue lines (win-k) are closer to the average of the regular blue lines,

i.e., average subsequence scores. Thus, we can conclude that the membership score  $f(x; M)$  computed by win-k acts as a better representative of the whole sample compared to min-k.

## 4 Experiments and Discussion

### 4.1 Experiment Setup

**Models.** We perform experiments with three model families: GPT-Neo [2], Pythia [1], and MobileLLM [7]. Since our work focuses on SLMs, we pick those models with  $\leq 1.5$ B parameters. We use the following models in our experiments: GPT-Neo 125M; Pythia 70M, 160M, 410M, 1B; MobileLLM 125M, 350M, 600M.

**Datasets.** We fine-tune SLMs on the following three datasets which are commonly used in the literature: WikiText, AGNews, and XSum. We created different versions of the datasets with different sample lengths:  $T = 32, 64$ , and  $128$ . We use  $T = 32$  by default, but report results with varying  $T$  in Section 4.4. To test MIA effectiveness, we construct balanced test sets that contain 350 members (used in fine-tuning) and 350 non-members (not used in fine-tuning).

**Fine-Tuning Parameters.** All models are fine-tuned using supervised fine-tuning (SFT) via the SFTTRAINER framework. The maximum sequence length is set to 2048 tokens, number of epochs is set to 2 (experiments are done with varying numbers of epochs in Section 4.4), batch size is set to 8, gradient accumulation is performed over 4 steps, and the learning rate is  $3 \times 10^{-5}$ .

**Attack Hyperparameters.** We compare win-k against attacks presented in Section 2.2. For the neighborhood attack, the number of neighbors is 100, and BERT is used as the masked language model for neighbor generation. For min-k and win-k, we experiment with varying  $k \in \{5\%, 10\%, 20\%, \dots, 90\%\}$  and  $w \in \{1, 2, 3, \dots, 10\}$ , and report the best results.

**Evaluation Metrics.** The effectiveness of MIAs is quantitatively evaluated using three metrics: AUROC (Area Under ROC Curve), TPR @ 1% FPR, and FPR @ 99% TPR.

### 4.2 Comparison with Existing MIAs

In this section, we compare win-k with existing MIAs to demonstrate its superior effectiveness. Table 1 contains the AUROCs of different MIAs under 8 different models and 3 fine-tuning datasets. In summary, Table 1 shows that win-k has the highest AUROC among all attacks in 17 out of 24 cases, demonstrating that win-k is generally more effective than the other attacks. We note that win-k outperforms the other MIAs more consistently especially when models are smaller, e.g., GPT-Neo 125M, Pythia 70M, and Pythia 160M. On larger models such as Pythia 410M or Pythia 1B, min-k can be tied with win-k, or min-k can surpass win-k by a small amount. This shows that win-k is indeed better for smaller language models. Another interesting observation is that win-k performs relatively worse on MobileLLM compared to GPT-Neo and Pythia families. A reason behind this could be the tokenizers. GPT-Neo and Pythia use similar tokenizers (GPT2Tokenizer and GPTNeoXTokenizer), both based on byte-pair

**Table 1.** AUROCs of different MIAs with varying models and datasets. MobLM is short for MobileLLM, Nbrhood is short for the Neighborhood attack, Lowercs is short for the Lowercase attack. The best attack in each case is highlighted in bold.

Dataset	Attack	GPT-Neo	Pythia	Pythia	Pythia	Pythia	MobLM	MobLM	MobLM
		125M	70M	160M	410M	1B	125M	350M	600M
WikiText	Nbrhood	67.0%	62.3%	61.8%	76.6%	83.6%	57.9%	59.9%	65.8%
	Lowercs	66.7%	65.6%	66.3%	81.1%	90.2%	59.2%	59.2%	68.7%
	Loss	74.4%	74.1%	77.1%	95.3%	98.5%	59.4%	60.2%	85.2%
	Zlib	73.6%	73.6%	76.9%	95.1%	98.4%	59.7%	59.8%	83.7%
	Min-k	76.0%	74.6%	77.6%	<b>96.0%</b>	<b>98.7%</b>	63.6%	63.9%	<b>85.3%</b>
	Win-k	<b>76.3%</b>	<b>75.1%</b>	<b>78.9%</b>	<b>96.0%</b>	98.4%	<b>65.1%</b>	<b>64.1%</b>	77.2%
AGNews	Nbrhood	78.3%	69.3%	68.4%	82.2%	91.0%	62.9%	65.3%	76.2%
	Lowercs	80.1%	71.7%	71.2%	84.2%	96.9%	65.0%	67.2%	88.2%
	Loss	85.1%	80.5%	79.4%	90.7%	98.1%	68.6%	70.6%	94.0%
	Zlib	83.6%	79.1%	78.0%	89.5%	97.4%	66.8%	69.1%	92.5%
	Min-k	86.6%	81.2%	81.8%	92.9%	98.3%	<b>76.8%</b>	<b>81.2%</b>	<b>94.2%</b>
	Win-k	<b>87.9%</b>	<b>83.4%</b>	<b>83.9%</b>	<b>93.2%</b>	<b>98.5%</b>	76.0%	79.4%	90.8%
XSum	Nbrhood	63.2%	61.8%	67.6%	77.5%	86.3%	57.1%	59.4%	67.2%
	Lowercs	64.7%	62.6%	68.3%	79.1%	89.6%	57.7%	60.2%	72.6%
	Loss	68.7%	69.9%	77.1%	90.6%	95.8%	59.4%	62.0%	80.6%
	Zlib	67.9%	69.1%	76.0%	89.5%	95.4%	59.1%	61.4%	78.1%
	Min-k	69.2%	69.9%	77.2%	<b>90.8%</b>	<b>95.9%</b>	60.2%	63.9%	<b>80.7%</b>
	Win-k	<b>69.9%</b>	<b>70.4%</b>	<b>78.0%</b>	<b>90.8%</b>	95.5%	<b>61.6%</b>	<b>64.8%</b>	75.1%

encoding and same vocabulary sizes (50,257 tokens). Yet, MobileLLM uses a Llama-based tokenizer for which the vocabulary size is 32,000.

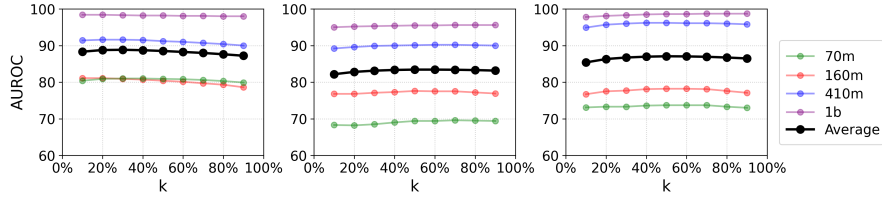
Next, we study the TPRs of the attacks @ 1% FPR. The results in Table 2 show that win-k is the best performing attack in this metric in 17 out of 24 cases. Win-k particularly emerges as the best performer on WikiText and AGNews. On the other hand, the Lowercase attack performs well on the XSum dataset. (If Lowercase did not exist, then win-k would have been the best-performing attack on XSum.) Based on all of our experiments, we observed that this exceptionally strong performance on Lowercase is limited to the strict setting of 1% FPR, e.g., Lowercase does not perform as well in terms of other metrics or at other FPR thresholds. Third, we study the FPRs of the attacks @ 99% TPR. Due to the page limit, we do not include the full table of results in the paper, but report that win-k has the best FPRs in 12 out of 24 cases. Considering there are 6 attacks under comparison, win-k is still the best-performing attack. However, its superiority is not as significant in this metric compared to the other two metrics.

### 4.3 Analysis of Win-k Hyperparameters

There are two main hyperparameters in win-k: window size  $w$  and fraction  $k$ . We report results with varying  $w$  between 2 and 10 in Table 3. For smaller models, e.g., less than 400M parameters, it can be observed that  $w = 2, 3$ , or 4 yield better AUROC in many cases. For example,  $w = 2$  and 3 typically perform the best on AGNews, and  $w = 3$  and 4 typically perform the best on XSum.

**Table 2.** TPR @ 1% FPR of different MIAs with varying models and datasets.

Dataset	Attack	GPT-Neo 125M	Pythia 70M	Pythia 160M	Pythia 410M	Pythia 1B	MobLM 125M	MobLM 350M	MobLM 600M
WikiText	Nbrhood	4.3%	4.0%	3.1%	10.3%	12.0%	0.6%	1.7%	1.7%
	Lowercs	4.6%	2.9%	5.1%	19.7%	55.7%	2.6%	2.6%	12.6%
	Loss	3.1%	2.6%	5.1%	24.9%	69.1%	1.1%	1.4%	13.4%
	Zlib	4.0%	4.0%	5.4%	27.7%	58.9%	0.9%	0.3%	13.4%
	Min-k	4.6%	4.6%	5.7%	30.3%	<b>75.4%</b>	4.6%	2.3%	<b>15.4%</b>
	Win-k	<b>6.3%</b>	<b>7.1%</b>	<b>7.1%</b>	<b>33.1%</b>	70.3%	<b>5.1%</b>	<b>4.0%</b>	<b>15.4%</b>
AGNews	Nbrhood	1.7%	2.9%	2.3%	3.4%	23.7%	0.9%	1.1%	6.0%
	Lowercs	6.0%	7.1%	5.4%	8.9%	27.4%	4.0%	5.1%	12.3%
	Loss	2.3%	4.0%	1.7%	4.6%	38.9%	1.1%	1.4%	14.6%
	Zlib	1.1%	1.1%	1.1%	1.4%	26.3%	1.1%	1.1%	2.0%
	Min-k	8.6%	9.4%	8.0%	12.6%	40.3%	3.4%	7.1%	15.1%
	Win-k	<b>16.6%</b>	<b>15.4%</b>	<b>12.3%</b>	<b>27.1%</b>	<b>64.6%</b>	<b>4.9%</b>	<b>7.7%</b>	<b>23.7%</b>
XSum	Nbrhood	4.0%	3.1%	5.7%	8.6%	15.4%	2.6%	2.9%	6.0%
	Lowercs	<b>4.6%</b>	<b>4.3%</b>	5.4%	4.3%	<b>25.1%</b>	2.6%	4.0%	<b>10.9%</b>
	Loss	0.3%	1.4%	6.6%	6.0%	18.0%	<b>4.6%</b>	2.9%	5.7%
	Zlib	1.7%	1.1%	2.9%	8.0%	15.4%	2.0%	2.3%	7.1%
	Min-k	0.6%	3.4%	5.7%	<b>11.4%</b>	<b>25.1%</b>	1.1%	3.7%	6.0%
	Win-k	2.0%	2.3%	<b>6.9%</b>	10.6%	17.7%	3.1%	<b>4.3%</b>	7.1%

**Fig. 4.** AUROCs of win-k with varying Pythia models and three datasets (left to right: AGNews, XSum, WikiText) under different  $k$ .

Yet, for larger models such as MobileLLM 600M and Pythia 1B, larger  $w$  are preferable, e.g., on both WikiText and XSum datasets,  $w = 8, 9$ , and  $10$  yield the highest AUROC. Overall, these results show that the best  $w$  is not fixed; it changes according to the size of the model. We also observe that if  $w$  is selected in parallel to this recommendation, the attack is not extremely sensitive to the precise value of  $w$ , since AUROCs in Table 3 vary by a moderate amount as  $w$  changes. Thus, it is sufficient to choose a good enough  $w$  following the above principle for win-k to perform well.

In Figure 4, we report results by varying the  $k$  parameter. To improve statistical significance, we repeat the experiment with multiple Pythia models with varying sizes (70M, 160M, 410M, 1B) and all three datasets. The average AUROCs across all models are shown in the plots, in addition to the AUROCs of each individual model. We observe from the plots that  $k$  values between 0.2 and 0.5 typically yield the highest AUROCs. Lower  $k$ , such as  $k = 0.2$  and  $0.3$ , are better on AGNews (reducing  $k$  to 0.1 yields lower AUROC). In contrast,  $k = 0.4$

**Table 3.** Impact of  $w$  on AUROCs of win-k under varying models and datasets.

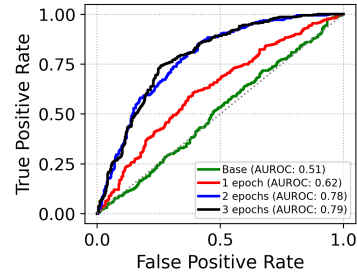
Dataset	Model	$w = 2$	$w = 3$	$w = 4$	$w = 5$	$w = 6$	$w = 7$	$w = 8$	$w = 9$	$w = 10$
WikiText	Pythia 70M	73.8%	74.1%	74.2%	74.3%	74.3%	74.2%	73.8%	73.7%	73.4%
	GPT-Neo 125M	62.5%	62.8%	63.1%	63.1%	63.0%	62.9%	62.7%	62.4%	62.2%
	Pythia 160M	77.4%	77.8%	77.9%	78.1%	78.2%	78.1%	77.8%	77.7%	77.5%
	MobLM 350M	61.7%	61.8%	61.7%	61.7%	61.3%	61.3%	61.1%	61.0%	61.2%
	Pythia 410M	95.0%	95.2%	95.4%	95.5%	95.6%	95.7%	95.8%	95.7%	95.7%
	MobLM 600M	76.0%	76.1%	76.6%	77.0%	77.0%	77.4%	77.8%	78.1%	78.8%
AGNews	Pythia 70M	82.3%	82.5%	82.1%	81.7%	81.2%	81.1%	80.9%	80.7%	80.6%
	GPT-Neo 125M	73.4%	73.4%	73.2%	73.0%	72.5%	72.0%	71.8%	71.5%	71.4%
	Pythia 160M	82.2%	82.4%	82.2%	81.8%	81.4%	81.1%	81.0%	80.6%	80.3%
	MobLM 350M	76.0%	75.6%	75.2%	74.8%	74.4%	74.0%	73.7%	73.4%	73.2%
	Pythia 410M	92.1%	92.1%	92.0%	91.9%	91.7%	91.5%	91.4%	91.3%	91.1%
	MobLM 600M	87.7%	87.7%	87.9%	88.1%	88.1%	88.2%	88.4%	88.7%	89.2%
XSum	Pythia 70M	98.1%	98.2%	98.2%	98.3%	98.2%	98.3%	98.3%	98.2%	98.2%
	GPT-Neo 125M	69.6%	69.9%	69.9%	69.5%	69.2%	68.8%	68.9%	68.9%	68.9%
	Pythia 160M	60.3%	60.6%	60.6%	60.5%	60.5%	60.4%	60.2%	60.1%	60.1%
	Pythia 160M	76.6%	77.1%	77.3%	77.2%	77.2%	77.0%	77.3%	77.1%	77.1%
	MobLM 350M	63.3%	63.2%	63.1%	63.1%	62.9%	62.9%	62.9%	62.9%	63.0%
	Pythia 410M	90.1%	90.4%	90.3%	90.2%	90.1%	90.0%	90.0%	89.9%	89.8%
	MobLM 600M	74.2%	74.3%	74.5%	74.9%	74.9%	75.1%	75.3%	75.7%	76.1%
	Pythia 1B	95.1%	95.2%	95.4%	95.4%	95.4%	95.4%	95.4%	95.4%	95.4%

and 0.5 work best on WikiText.  $k = 0.4$  works best on XSum as well; however, XSum shows the smallest change in AUROCs as  $k$  changes. Overall, we observe the trend that  $k$  should be selected neither too small nor too large. To achieve the best results, we recommend  $k$  between 0.3 and 0.5.

#### 4.4 Impact of Data and Fine-Tuning Related Parameters

Finally, we investigate the impacts of parameters related to text samples and fine-tuning. In Figure 5, we fine-tune Pythia 160M using XSum for varying numbers of epochs: 0 epochs (base model), 1, 2, and 3 epochs. Executing win-k on the base model (0 epochs) indeed yields an AUROC close to 0.5, i.e., random guess. As we increase the number of epochs, AUROCs increase. There is a substantial increase from 0 epochs to 1 epoch, and also from 1 epoch to 2 epochs. However, the amount of increase from 2 epochs to 3 epochs is not very large, which shows that the model’s susceptibility becomes saturated. Overall, it is intuitive that increasing the number of epochs increases model susceptibility, since the log probabilities produced by the model become more dominated by the fine-tuning dataset. It is important to note that an SLM like Pythia 160M becomes quickly vulnerable to win-k, even with 1 or 2 epochs of fine-tuning.

In Table 4, we investigate how the size of the text samples impacts attack effectiveness. We vary the value of  $T$  by taking long samples (i.e.,  $T \geq 128$ ) and truncating them to  $T = 32, 64$ , and 128. We perform the experiment using

**Fig. 5.** Impact of changing number of epochs in terms of AUROC.



**Table 4.** AUROCs of min-k and win-k with varying models and datasets under different number of tokens  $T$ .

Dataset	Tokens	GPT-Neo 125M		Pythia 160M		Pythia 410M	
		Min-k	Win-k	Min-k	Win-k	Min-k	Win-k
WikiText	$T = 32$	76.0%	71.7%	77.6%	78.9%	96.0%	96.0%
	$T = 64$	83.1%	83.5%	84.8%	86.3%	98.8%	98.9%
	$T = 128$	87.2%	87.0%	84.8%	86.3%	99.3%	99.5%
XSum	$T = 32$	69.2%	69.9%	77.2%	78.0%	90.8%	90.8%
	$T = 64$	70.9%	71.9%	79.0%	80.0%	94.3%	94.8%
	$T = 128$	76.9%	78.5%	84.2%	86.2%	97.6%	97.8%

three models (GPT-Neo 125M, Pythia 160M, and Pythia 410M), two datasets (WikiText and XSum), and two attacks. As the results in Table 4 show, increasing  $T$  typically yields a substantial increase in AUROC. The amount of increase is more noticeable in smaller models like GPT-Neo 125M and Pythia 160M. In contrast, the AUROCs in Pythia 410M are already high when  $T = 32$ ; thus, the amount of increase from  $T = 32$  to 64 and 128 is less noticeable.

## 5 Discussion and Conclusion

**Summary.** SLMs are rapidly gaining traction as efficient and deployable alternatives to LLMs in resource-constrained and on-device AI applications. In this paper, we examined the vulnerability of SLMs to MIAs. Our analysis revealed that the effectiveness of MIAs declines as model size decreases. We therefore proposed win-k, a new MIA which generalizes the min-k attack by aggregating log probability scores over sliding windows of tokens. Experiments on eight SLMs across three datasets and three evaluation metrics showed that win-k outperforms prior attacks, especially on smaller models. Furthermore, we provided practical insights into the selection of win-k’s hyperparameters.

**Limitations and future work.** First, our current work is limited to SLMs with  $\leq 1$ B parameters. Generalizing to larger models (e.g., up to 5B or 7B parameters) would be a valuable future work direction. However, full fine-tuning of such models using SFTTRAINER is unlikely to be feasible, hence PEFT methods such as LoRA may be needed. For consistency, we stick with SFTTRAINER in the paper and leave fine-tuning of larger models with LoRA to future work. Second, it would be interesting to consider issues arising from the number of member versus non-member samples as well as the types of samples (e.g., scientific texts versus news articles). Attacks targeting specific contexts or targeted content may be considered. Third, while win-k relies on a sliding mean of log probabilities, it does not examine alternative aggregation strategies (e.g., median, trimmed mean, max pooling). We will consider extending our attack with such aggregation strategies. Finally, we will study defenses against win-k and MIAs in general. As MIAs exploit models’ tendency to overfit, one mitigation strategy could be regularization (dropout, L1 or L2 regularization). Noise can be introduced to log probabilities to mask membership or differentially private fine-tuning methods can be utilized as defenses.

**Acknowledgements** This study was supported by The Scientific and Technological Research Council of Türkiye (TUBITAK) under grants numbered 123E179 and 125E059. The authors thank TUBITAK for their support.

## References

1. Biderman, S., Schoelkopf, H., Anthony, Q.G., Bradley, H., O'Brien, K., Hallahan, E., Khan, M.A., Purohit, S., Prashanth, U.S., Raff, E., et al.: Pythia: A suite for analyzing large language models across training and scaling. In: International Conference on Machine Learning. pp. 2397–2430. PMLR (2023)
2. Black, S., Gao, L., Wang, P., Leahy, C., Biderman, S.: GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow (2021). <https://doi.org/10.5281/zenodo.5297715>
3. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Nee-lakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. *Advances in Neural Information Processing Systems* **33**, 1877–1901 (2020)
4. Carlini, N., Tramer, F., Wallace, E., Jagielski, M., Herbert-Voss, A., Lee, K., Roberts, A., Brown, T., Song, D., Erlingsson, U., et al.: Extracting training data from large language models. In: 30th USENIX Security Symposium. pp. 2633–2650 (2021)
5. Duan, M., Suri, A., Mireshghallah, N., Min, S., Shi, W., Zettlemoyer, L., Tsvetkov, Y., Choi, Y., Evans, D., Hajishirzi, H.: Do membership inference attacks work on large language models? *arXiv preprint arXiv:2402.07841* (2024)
6. Hu, S., Tu, Y., Han, X., He, C., Cui, G., Long, X., Zheng, Z., Fang, Y., Huang, Y., Zhao, W., et al.: Minicpm: Unveiling the potential of small language models with scalable training strategies. *arXiv preprint arXiv:2404.06395* (2024)
7. Liu, Z., Zhao, C., Iandola, F., Lai, C., Tian, Y., Fedorov, I., Xiong, Y., Chang, E., Shi, Y., Krishnamoorthi, R., et al.: Mobilellm: Optimizing sub-billion parameter language models for on-device use cases. In: International Conference on Machine Learning (2024)
8. Lu, Z., Li, X., Cai, D., Yi, R., Liu, F., Zhang, X., Lane, N.D., Xu, M.: Small language models: Survey, measurements, and insights. *arXiv preprint arXiv:2409.15790* (2024)
9. Mattern, J., Mireshghallah, F., Jin, Z., Schoelkopf, B., Sachan, M., Berg-Kirkpatrick, T.: Membership inference attacks against language models via neighbourhood comparison. In: The 61st Annual Meeting Of The Association For Computational Linguistics (2023)
10. Shi, W., Ajith, A., Xia, M., Huang, Y., Liu, D., Blevins, T., Chen, D., Zettlemoyer, L.: Detecting pretraining data from large language models. In: 12th International Conference on Learning Representations, ICLR (2024)
11. Shokri, R., Stronati, M., Song, C., Shmatikov, V.: Membership inference attacks against machine learning models. In: IEEE Symposium on Security and Privacy (SP). pp. 3–18. IEEE (2017)
12. Truex, S., Liu, L., Gursoy, M.E., Yu, L., Wei, W.: Demystifying membership inference attacks in machine learning as a service. *IEEE Transactions on Services Computing* **14**(6), 2073–2089 (2019)
13. Yeom, S., Giacomelli, I., Fredrikson, M., Jha, S.: Privacy risk in machine learning: Analyzing the connection to overfitting. In: 31st Computer Security Foundations Symposium (CSF). pp. 268–282. IEEE (2018)

## **Session 2: Applied Cryptography & Statistics**

# Advanced Electronic Signatures and GDPR: Reconciling the Concepts

Paweł Kostkiewicz<sup>1</sup>[0000–0003–4857–8787],  
Mirosław Kutylowski<sup>1</sup>[0000–0003–3192–2430], and  
Gabriel Wechta<sup>1</sup>[0009–0009–8560–5300]

NASK National Research Institute, Warsaw, Poland  
{pawel.kostkiewicz, miroslaw.kutylowski, gabriel.wechta}@nask.pl

**Abstract.** Digital signature schemes developed by the cryptographic community have been adopted as *advanced electronic signatures* in the legal framework of many countries, including the European Union. In their current implementations, certification practices, and legal practice, electronic signatures are orthogonal to privacy protection. In fact, protecting data origin and integrity with advanced electronic signatures creates many challenges from the point of view of GDPR.

In this paper, we show that conflicts between the legal concept of electronic signatures and the strength of digital signatures on one side, and the paradigms of privacy-by-design, data minimization, etc. can be resolved by slightly reshaping the signature schemes and reinterpreting certain legal concepts.

**Keywords:** GDPR, eIDAS, Advanced Electronic Signature, Hash Function, Merkle Tree

## 1 Introduction

Digital signatures provide essential cryptographic guarantees: authentication, integrity, and non-repudiation, which make them valuable in many digital application areas. However, the standard approach to creating a signature, that is, treating the message as a single array of bytes, can sometimes become a limitation. This rigid model may prevent certain desirable use cases that are common in the physical world but difficult to replicate digitally. For example, with a handwritten signature on a physical document, one can partially cover the page to hide sensitive information while still proving that the visible portions are authentic. In the digital realm, achieving a similar form of *selective disclosure* is not straightforward under conventional signature schemes.

In this paper, we examine this problem in detail, place it in the context of current legal frameworks (namely, eIDAS [29,12] and GDPR [30]), and present two straightforward schemes that can be used with any existing standard cryptographic primitives. In particular, our approach can be used with widely standardized algorithms, and thus be inline with legislation such as eIDAS. Importantly, it does not introduce an additional burden on end users as it does not alter the

existing interfaces (meaning that creation and verification of signature from the user's point of view remains the same), except introducing a new functionality called selective disclosure.

### 1.1 Traditional Model of Digital Signatures

The first challenge is that an electronic signature scheme applied in practice should be able to create a signature for a document of arbitrary size. Thus, it cannot be assumed that a message is an element of a fixed algebraic structure. This practically eliminates schemes (e.g., [20,5]) that are not based on a cryptographic digest (in practice, a hash function and the Random Oracle Model [2]).

Another common practical issue is that signatures often need to be generated using a secure signature creation device (cf. WSCD in eIDAS [12]). In many cases, this device is a special-purpose component (e.g., a smart card), with limited storage capacity and/or low-bandwidth communication capabilities. In this case, uploading an entire document to the device is impractical. Instead, a common strategy is to upload only a hash of the document, or a small portion of it together with an intermediate hash [26]; see also p. 465 in [31]. Analogously, for sponge hash functions [3], the signing device may execute only the last absorbing step. To alleviate the problems mentioned above, the common approach is a two-stage signing process for a document  $D$ :

1. Run some algebraic precomputation to generate an element  $r$ .
2. Create a cryptographic digest  $h$  of  $D$  and the precomputed value  $r$  using hash function  $\text{Hash}$ , concretely  $h := \text{Hash}(D, r)$ .
3. Apply the core signing procedure for  $h$ , using the secret signing key  $\text{sk}$ , that is  $\sigma := \phi(h, \text{sk})$ , where  $\phi$  is some algebraic operation based on a difficult cryptographic problem.

During the verification procedure, the values  $h$  and  $r$  are recalculated in the first step. After that, an algebraic test is executed. It involves  $\sigma$ ,  $h$  and the public key  $\text{pk}$  corresponding to  $\text{sk}$ . Note that this approach is followed by major digital signature schemes.

An immediate consequence of this approach is that any benign (e.g., privacy-preserving) change to the document before presenting it for verification (such as replacing personal identification data with a pseudonym) will produce a different value of  $h$ . Consequently, the signature on the document is no longer valid, and its entire proof value is lost. To address this problem we employ a document-structure-aware hash function (details are provided in Section 3).

In the next section, we discuss several practical situations and show that, in these cases, the observation made above is the Achilles' heel of the concept of electronic signature.

### 1.2 Privacy Protection Versus Digital Signatures

Although advanced electronic signatures provide very strong arguments for data origin and integrity, in certain situations, their strength becomes a significant

problem severely limiting their application scope. In this section, we highlight a variety of scenarios in which reconsidering the standard approach to digital signatures could provide substantial benefits.

*Partial disclosure of signed documents.* In certain scenarios, such as public administration procedures, legal documents are signed and subsequently published to comply with the right of access to public information. However, these documents may contain information that must remain confidential. Examples include personal data protected according to the GDPR, information related to public security, or other categories of data protected by law. A concrete example is the publication of court judgments in Poland, which must be anonymized in the sense that all personal data contained in court decisions must be removed<sup>1</sup> (except for the names of court personnel and judges involved). Such anonymized court decisions are available online.<sup>2</sup>

In this case, the original digital signature on the document cannot be used directly to check the blinded document. Even worse, if the scope of blinding is limited, a brute-force attack may be used to reconstruct the original document — the signature serves as an oracle for checking the correctness of a guess.

For this reason, if certain parts of a document need to be blinded, the document should be re-signed. However, this introduces not only additional effort but also new risks, as, for example, a new signature might be applied to content that differs from the original in the non-blinded parts.

*Personalized disclosure.* It may happen that a document  $D$  contains a wide range of data ( $D$  could be a complicated contract and/or an extensive technical documentation), where an individual reader should read only its selected parts according to the data minimization paradigm.

The classical approach to dealing with this problem is to split the document into separate parts and sign each part separately. This solution might be quite tedious, as splitting the document must occur in advance, while the situation may dynamically change, especially if the subject covered by the  $D$  is complicated.

*Electronic document management systems.* Electronic flow of documents may require complicated access rights to document contents. For example, a reviewer in PhD proceedings in Poland submits a standard bill containing the information such as bank account, place of residence, personal identification number, and other data required by the tax authorities. However, the bill must be accepted by a person responsible for verifying the PhD report submitted by the reviewer. In this case, a staff member of the entity granting the degrees gets access to data such as bank account number of the reviewer. This data is unnecessary for payment approval. This directly violates the data minimization paradigm from GDPR as well as general cybersecurity rules.

In electronic document management systems, we require flexible rules for data access. Moreover, they should be easy to handle from the user's point

<sup>1</sup> An operation known as *blinding*.

<sup>2</sup> See <https://orzeczenia.ms.gov.pl>.

of view. The ultimate target would be to achieve the level of flexibility and expressivity achieved by the best access control systems. Reaching this goal with the current electronic signature model seems infeasible.

*Right to be forgotten.* A signed document  $D$  may contain personal data that should be erased at the legitimate request of a data subject  $A$  [30]. However, the document  $D$  may contain data that must be retained, in particular, due to some legal obligation. In this case, we have a deadlock: blinding certain parts of  $D$  makes the signature contained in  $D$  invalid and violates the obligation to keep  $D$  in a verifiable form. On the other hand, without blinding the rights of the data subject  $A$  granted by GDPR are violated.

*Hiding signatory.* In certain situations, the information on the signatory should be protected. It may concern documents that are processed within an organization with multiple persons involved, and signing them to mark acceptance. When the document leaves the organization, only a few chosen signatures should be attached to the document's text. However, during the intermediate stages of document processing, the next signature is created for the original document appended with the previous signatures. In this case, it might be challenging to remove some of the intermediate signatures.

A similar problem may concern public key certificates (which again are secured with an electronic seal). Not all fields of the certificate should be visible to every Verifier — an example is a certificate for personal signature created by personal identity cards in Poland, which includes the signatory's personal identification number PESEL. This number is not secret; however, due to the threat of identity theft, it should not be distributed unless necessary. In most cases, it is not.

### 1.3 Related Work

So far, selective disclosure has been explored primarily in the Self-Sovereign Identity domain, particularly in the context of privacy-preserving credential presentation [4,15,7]. In parallel, recent European legislative developments have significantly influenced the design of digital services. Key regulations include the eIDAS framework [29,12], the GDPR [30], and the Whistleblowing Directive [13]. These initiatives set the legal foundation for trust, interoperability, and privacy in the European digital ecosystem, providing both opportunities and constraints for the implementation of selective disclosure technologies. However, introduction of new cryptographic techniques for legal use cases is guarded by a list of scrutinized standards (see, e.g., [9,23,17,22]).

## 2 Digital Signatures According to eIDAS

According to the eIDAS Regulation [12,29], users of European Digital Identity Wallets (EDIW) should be able to create and use electronic signatures that are

accepted across the EU (see Recitals 19 and 20). eIDAS sets out the criteria for an advanced electronic signature in Article 26. These requirements, while not framed in standard cryptographic language, align closely with traditional cryptographic signature properties (with one exception). Specifically, an advanced electronic signature must be

- (a) uniquely linked to the signatory and (b) capable of identifying the signatory — both aspects relate to *non-repudiation*,
- (c) created using electronic signature creation data that the signatory can, with a high level of confidence, use under his sole control — *authentication*, and
- (d) **linked to the signed data in such a way that any subsequent changes are detectable** — which goes beyond the traditional definition of *data integrity*. Conventionally, any modification to the signed data typically renders the signature invalid. However, under the eIDAS framework, one might interpret this requirement in two ways:
  - Minimum: Any alteration of the data makes the signature invalid.
  - Maximum: It is possible to detect precisely what was changed; signatures covering the modified parts and the overall document are invalid, but the unchanged parts can still be verified correctly.

In our opinion, the second interpretation is not only more pragmatic, but also follows the standard interpretation of legal norms. Indeed, for the first interpretation (minimum), the wording of the legal text that more closely reflects the interpretation would be “...linked to the signed data in such a way that any subsequent changes *make the signature invalid*.”

## 2.1 ETSI Recommended Digital Signatures Algorithms

The general legal definitions should be confronted with practice, where technical recommendations and standards play the crucial role. Among others, any deviation from the recommendations and standards creates multiple problems, ranging from limited availability on the market to substantially harder certification and market acceptance. Let us briefly discuss ETSI TS 119 312 V1.4.3 (2023-08) [9], the official EU list of algorithms to be used for electronic signatures. It recommends the following signature algorithms:

**RSA:** As specified in RFC 8017 [22], the RSA-PKCS1-v1\_5 and RSA-PSS schemes begin signing by applying encoding procedures (EMSA-PKCS1-v1\_5 and EMSA-PSS, respectively) to a hash of the message.

**ECDSA:** As defined in FIPS 186-5 [23], the message is first processed using an approved hash function or an extendable-output function.

**ECGDA:** According to ISO/IEC 14888-3 [17], this variant of ECDSA also begins by hashing the message to be signed.

In all cases,<sup>3</sup> the entire document is hashed, so any modification is detectable and results in the signature’s invalidation.

<sup>3</sup> The DSA is omitted here, as it is no longer approved for signature generation according to FIPS 186-5 [23].



## 2.2 Presentation of Attributes

The eIDAS regulation concerns the use of electronic signatures in two distinct contexts. First, eIDAS mandates that EDIW users must be able to create and use electronic signatures. These signatures must be accepted across the EU “... by default and free of charge, without having to go through any additional administrative procedures.” (ref. Recital 19). One key use case is the ability to “... sign or seal self-claimed assertions or attributes ...” — a privacy-preserving mechanism that allows users to reveal only specific parts of an identity document, rather than the entire document.

Second, electronic signatures also play a central role within the eIDAS trust service framework in attribute presentation, one of the key EDIWs’ functionalities. Specifically, qualified electronic attestations of attributes (Article 45d) and electronic attestations issued by or on behalf of public sector bodies responsible for authentic sources (Article 45f) must comply with the requirements outlined in Annexes V and VII, respectively. In both cases, a qualified electronic signature from the issuing qualified trust service provider or public authority is required.

However, this collides with Article 5a.4(a) that requires that the EDIW must enable users to “request, obtain, select, combine, store, delete, share, and present ... personal identification data and, where applicable, in combination with electronic attestations of attributes ... while ensuring that *selective disclosure* of data is possible.” Clearly, to support selective disclosure, signature verification mechanisms must allow for fine-grained validation without requiring the user to present the entire attestation.

Finally, to further support the compatibility of eIDAS with our proposed reconciled approach to signature mechanisms, it is worth noting that the Commission Decision 2015/1506 [11], which specifies formats for advanced electronic signatures and seals under eIDAS, endorses XML,<sup>4</sup> CMS and PDF formats.

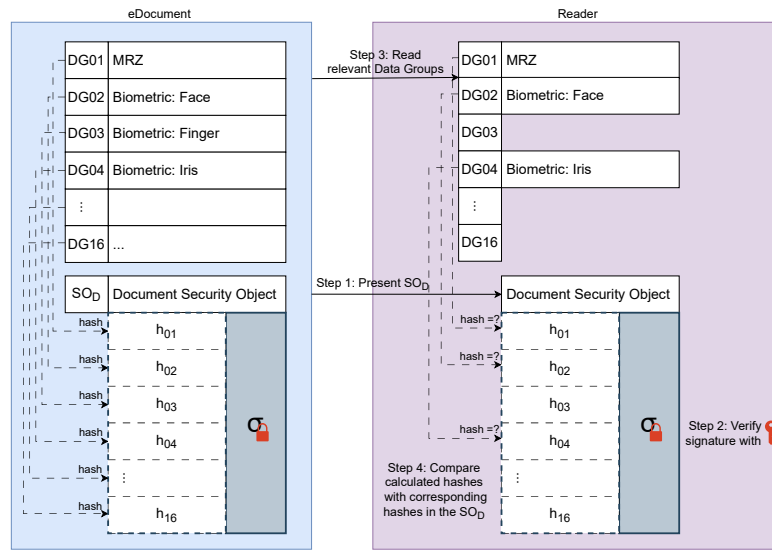
## 2.3 ICAO

It should be noted that a similar to ours, but less general approach has been adopted by the ICAO [16] for authenticating the personal data of holders of travel eDocuments. These eDocuments (e.g., passports, personal identity cards) contain both standard identification data (such as name, date of birth, etc.) and sensitive biometric data (such as fingerprints, iris scans, etc.). Access to the former is available to anyone holding a travel eDocument, for instance, by using an e-passport reader app such as [28].

An eDocument holds 16 Data Groups, which store identification data, and a Document Security Object  $SO_D$ . During the preparation phase, the Document Issuer computes the hash of each Data Group and signs the collection using their Document Issuer Public Key (represented by the red lock in Figure 1). Later, in the presentation phase, the eDocument provides  $SO_D$  to the Reader.

<sup>4</sup> Among these, XML is especially well-suited for implementing structures that enable granular access to specific fields in attestations.

Upon receiving it, the Reader verifies the signature in  $SO_D$  using the Document Issuer's Public Key (the red key in Figure 1). If the verification is successful, the Reader requests the relevant Data Groups from the eDocument. Once received, the Reader compares their hashes with those included in  $SO_D$ . If they match, the Reader accepts the data as authentic. In the described mechanism, the Document Issuer signs the root of a flat Merkle tree [21], where the leaves correspond to the individual Data Groups.



**Fig. 1.** Simple version of *passive authentication* of [16].  $\sigma$  denotes Document Issuer's signature on hash collection. For clarity, we omit both the validation of the Document Issuer's certificate and the authentication of the Reader to the eDocument prior to presentation.

### 3 Signatures with Selective Document Disclosure

The central conceptual shift that we propose is to enable selective disclosure of a signed document  $D$ , while preserving the ability to validate the signature as if it were applied to the complete original version of  $D$ .<sup>5</sup> Formally, we define *structured signature* scheme  $\text{SSign}$  built on top of a standard signature scheme  $\text{Sign}$  with the following procedures:

**Key generation:** Identical to the standard key generation procedure in  $\text{Sign}$ ; that is, generate a key pair  $(\text{sk}, \text{pk})$ .

<sup>5</sup> A similar concept, but in terms of files (not documents), can be found in ETSI standard [14, ref. Associated Signature Containers].

**Signature creation:** Instead of applying the standard hash function `Hash` to obtain the digest of a message  $D$  before invoking `Sign`, the scheme computes a structured hash  $h := \text{SHash}(D)$ , and then applies `Sign`. The resulting signature is denoted  $\sigma$ .

**Selective disclosure:** The signer or anyone who knows  $D$  and  $h$ , can produce a blinded version  $D'$  of the original document  $D$ .

**Signature verification:** The input consists of the blinded document  $D'$ , the original signature  $\sigma$  on the full document  $D$ , and the public key `pk`. Verification succeeds if and only if:

1. there exists a document  $D$  from which  $D'$  can be obtained via selective disclosure, and
2.  $\sigma$  is a valid signature on  $D$  under the public key `pk`.

The selective disclosure procedure defined this way should be iterative that is, it can be applied not only to the original document  $D$ , but also to any derived document  $D'$  obtained through selective disclosure. In particular, one may derive an even more restricted document  $D''$  from  $D'$  and continue this process iteratively. Concrete instantiations of `SSign` are presented in Sections 3.2 and 3.3.

### 3.1 Document with DAG Structure

The first step towards defining `SHash` is finding a graph structure of a document:

- In public and private administration, XML documents are frequently used, in particular with a predefined standard structure (see Figure 2). For example, the Polish Ministry of Finance publishes a catalog of tax documents.<sup>6</sup> Each XML document has the explicit structure of a tree, with document data items having well-defined locations and types. For XML documents, there is no need to convert to a graph representation, as it is explicit.
- The documents generated by humans, say in English, consist of sentences, which in turn form paragraphs. The paragraphs may form sections, etc. This relatively flat tree structure relates to the semantic composition of the text and can be explicitly created by the text author.
- A text file is a sequence of characters, but it can be analyzed by automatic Natural Language Processing (NLP) tools to determine its structure. The essential part of such analysis is to convert the text into a sequence of tokens and then to find the structures of how the tokens are composed. A clear advantage of this approach is that a token, such as an identifier of an individual, occurring in different places is treated as the same object (see Figure 3). The above conversion can be done by external LLM tools [1,24] (with all concerns related to data privacy); however, there are open-access LLMs that can be fine-tuned for this specific use case and run locally (e.g., DeepSeek R1 Offline [8]).

<sup>6</sup> See <https://www.podatki.gov.pl/e-deklaracje/dokumentacja-it/struktury-dokumentow-xml>.

```

<xsd:element name="PozycjeSzczegolowe">
  <xsd:complexType>
    <xsd:sequence>
      <xsd:annotation>
        <xsd:documentation>
          1. Należności ze stosunku: pracy, służbowego, spółdzielczego i z ...
        </xsd:documentation>
      </xsd:annotation>
      <xsd:element name="P_29" type="TKwota2Nieujemna">
        <xsd:annotation>
          <xsd:documentation>Przychód</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      ...
      <xsd:element name="P_30" type="TKwota2Nieujemna" minOccurs="0">
        <xsd:annotation>
          <xsd:documentation>Koszty uzyskania przychodów</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      <xsd:element name="P_31" type="TKwota2Nieujemna">
        <xsd:annotation>
          <xsd:documentation>Dochód</xsd:documentation>
        </xsd:annotation>
      </xsd:element>
      ...
    </xsd:sequence>
  </xsd:complexType>
</xsd:element>

```

POLTAX POLA JASNE WYPEŁNIA PŁATNIK, POLA CIEMNE WYPEŁNIA URZĄD SKARBOWY. WYPEŁNIAC NA MASZYNIE, KOMPUTEROWO LUB RĘCZNIE, DUŻYMI, DRUKOWANYMI LITERAMI, CZARNYM LUB NIEBIESZYM KOLOREM. Składanie w wersji elektronicznej: [www.portalpodatkowy.mf.gov.pl](http://www.portalpodatkowy.mf.gov.pl)

E. DOCHODY PODATNIKA, POBRANE ZALICZKI ORAZ POBRANE SKŁADKI <sup>9)</sup>					
Źródła przychodów	Przychód <sup>7)</sup>	Koszty uzyskania przychodów <sup>8)</sup>	Dochód (b - c)	Dochód zwolniony od podatku <sup>7)</sup>	Zaliczka pobrana przez płatnika
	zł. gr.	zł. gr.	zł. gr.	zł. gr.	zł.
1. Należności ze stosunku: pracy, służbowego, spółdzielczego i z pracy nakładczej, a także zasiłki pieniężne z ubezpieczenia społecznego wypłacone przez zakład pracy, o którym mowa w art. 31 ustawy, oraz płatników, o których mowa w art. 42e ust. 1 ustawy	29.	30.	31.	32.	33.
W poz. 34 należy wykazać przychody, do których zastosowano odliczenie kosztów uzyskania przychodów na podstawie art. 22 ust. 9 pkt 3 ustawy.	34.	35.			
9) Miesięczność w tabeli z odliczeniami w odliczeniu	36.		37.		38.

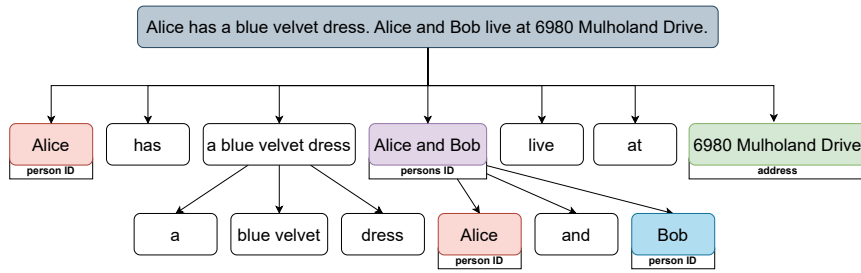
**Fig. 2.** Excerpt from the official Personal Income Tax XML file (PIT-11(16)\_v1-0.xsd, top) and the corresponding PDF form (bottom), both sourced from the Polish Ministry of Finance website.

It follows that with a limited effort, we can represent any document  $D$ , whether through an XML structure or tokenization,<sup>7</sup> by a Directed Acyclic Graph (DAG). We shall use  $\text{DAG}(D)$  to denote DAG based on document  $D$ , where:

- the nodes of  $\text{DAG}(D)$  are labeled and have unique identifiers ID,
- the content of  $D$  is encoded into labels of certain nodes (possibly not only the leaves),

<sup>7</sup> Note that we do not commit to a single “best” method, as the most suitable approach depends heavily on the specific scenario. For example, if a document is created with selective disclosure in mind, representing it as XML introduces minimal overhead for the creator — especially when issuing many copies with the same structure. For documents without such a structure, simple parsers may suffice. A solution based on LLMs that retrieves the structure of a document requires more consideration (see Section 4) but can still be applied effectively in ad hoc cases.

- the structure of  $\text{DAG}(D)$  reflects the semantic structure of document  $D$ ,
- it can be assumed that there is a single maximal (root) element in  $\text{DAG}(D)$  (having indegree zero); if there is more than one maximal element, we create an extra root node of  $\text{DAG}(D)$ , to which we connect all maximal nodes.



**Fig. 3.** An example of representing  $D$  as a DAG, with certain nodes labeled as potentially private data. Note that some nodes contain more than one word; this is why we use a more complex DAG representation rather than a simple flat tree. This choice was deliberate: 1) Context may reveal auxiliary information. Consider the term “Alice and Bob”. Assume for a moment that we do not have the term “Alice and Bob” as a single unit, but instead have three separate terms: “Alice”, “and”, and “Bob”. If, for example, Bob is the target of pseudonymization  $D$  and Alice is not, then blinding only the term “Bob” would be insufficient, as it would reveal the fact that the pseudonymized individual lives with Alice. 2) Certain languages (e.g., Chinese) may require different approach for semantic decomposition than English.

An instantiation of  $\text{DAG}(D)$  (see Figure 3 for an example) depends on the strategy to convert a text document  $D$  to a graph representation. For XML documents, it is immediate, while, say, for PDF documents, it might be a moderate standardization challenge due to multiple design decisions.

*Selective disclosure of a document.* If a document  $D$  with DAG structure  $\text{DAG}(D)$  has to be revealed selectively, then we blind a chosen part of the graph  $\text{DAG}(D)$  — if a node  $A$  is blinded, then all its successors are blinded as well.

To enable fluent human-machine interaction, it is essential to develop tools that convert the blinded  $\text{DAG}(D)$  into a human-readable format. Base libraries are available in most programming languages (for example, [25]).

*Example 1.* In XML tax document, we can selectively eliminate certain field values (like the taxpayer’s ID number) or blind an entire subtree (e.g., corresponding to the declared income). In the latter case, we get a document that witnesses that a given person has an active taxpayer status, while the confidential data regarding the income is removed. Such a blinded document might

useful, for example, in some towns, municipal authorities offer discounts on public transportation for their residents, provided that the taxpayer declares this municipality in the tax declaration.

*Example 2.* In case of a text document  $D$  converted to  $\text{DAG}(D)$  with an NLP tool, the resulting graphs have tokens corresponding to nodes with no incoming arcs. Some tokens correspond to physical persons identifiers or to data enabling identification of a data subject (e.g., a token being a part of the address). Note that such a token may correspond to multiple occurrences in the original text; note that manual anonymization may overlook some appearances. With a  $\text{DAG}(D)$ , it suffices to eliminate a selected token to eliminate all of its appearances.

In Sections 3.2 and 3.3, we present two approaches to creating signatures that enable partial disclosure of a document by utilizing its semantic DAG structure.

### 3.2 Merkle Tree Approach

*Preliminary definitions.* In this section, we consider a  $\text{DAG}(D)$  where the order of children of each node is defined and there is a single root - the only node with indegree 0 (recall that we consider directed edges that point in the direction from the root to the leaves). In particular, it does not need to be a tree.

Each node  $A$  of the tree may correspond to a part of the document  $D$  according to a relevant semantic structure. For the sake of generality, we assume that the text of  $D$  is spread not only among leaf nodes of  $\text{DAG}(D)$ , but possibly also among the non-leaf nodes. Let  $\text{ID}_A$  be the identifier of node  $A$ , and let  $T(A)$  denote its label. The label  $T(A)$  is either a text from  $D$  or an artifact of  $\text{DAG}(D)$  creation. If  $T(A)$  is a part of the text  $D$ , then we associate with  $A$  a random bit-string (salt)  $T_{\text{salt}}(A)$ , which is long enough to make brute-force preimage attacks against a hash function infeasible. Define

$$T^+(A) = \begin{cases} (T(A), T_{\text{salt}}(A)), & \text{if } T_{\text{salt}}(A) \text{ is defined,} \\ T(A), & \text{otherwise.} \end{cases}$$

*Computing SHash.* The hash value  $h(A)$  of a node  $A$  is defined recursively by the following formula:

$$h(A) = \begin{cases} \text{Hash}(T^+(A)), & \text{if } A \text{ is a leaf (outdegree zero),} \\ \text{Hash}(\text{Hash}(T^+(A)), h(A_1), \dots, h(A_u)), & \text{otherwise,} \end{cases}$$

where  $A_1, \dots, A_u$  are the child nodes of  $A$ . The values of  $h$  are calculated bottom-up starting from the leaves of  $\text{DAG}(D)$ .

Let  $h_{\text{root}}$  denote the hash value at the root node of  $\text{DAG}(D)$ . The final output is  $(h_{\text{root}}, \{(\text{ID}_{A_i}, T_{\text{salt}}(A_i))_i\})$  for all nodes  $A_i$  where  $T_{\text{salt}}(A_i)$  was defined.

*Signature creation.* The signing algorithm **Sign** is applied to produce the signature  $\sigma$ . The only difference from the standard procedure is that the hash value  $h = \text{Hash}(D)$  is replaced by  $h := h_{\text{root}}$  returned by **SHash**( $D$ ).

*Selective disclosure.* A User holds document  $D$  (and thereby also  $\text{DAG}(D)$ ), a signature  $\sigma$ , and the salt values  $T_{\text{salt}}(A_i)$ . The following steps are executed:

1. Construct the *blinded*  $\text{DAG}(D')$  by selecting a subset of nodes from  $\text{DAG}(D)$  whose label is to be hidden (we shall call them *blinded nodes*).
2. Replace  $T(A)$  of each blinded node  $A$  with the empty symbol  $\blacksquare$ .
3. Create a list  $L$  that enables reconstruction of  $h_{\text{root}}$ :
  - If node  $A$  is blinded, insert  $(\text{ID}_A, \text{Hash}(T^+(A), 0))$  into  $L$ .
  - If node  $A$  is non-blinded, insert  $(\text{ID}_A, T_{\text{salt}}(A), 1)$  into  $L$ .
4. Output  $(\text{DAG}(D'), \sigma, L)$  to the Recipient of the selectively disclosed signature.

*Signature verification for selectively disclosed  $D$ .* The specific part of signature verification is the recalculation of  $h_{\text{root}}$ . It is easy to see that with  $L$ , the Verifier can recompute  $h_{\text{root}}$ : for each non-blinded node  $A$ ,  $\text{Hash}(T^+(A))$  must be computed, using the part  $T(A)$  of the document  $D$ . For a blinded node  $B$ , the Verifier uses  $\text{Hash}(T^+(B))$  from the list  $L$  and cannot derive  $T(B)$  due to the application of the (unknown) salt string.

### Discussion.

*Note 1.* The text document  $D'$  corresponding to  $\text{DAG}(D')$  can be visualized in a manner analogous to the so called “black box” redaction of classified documents, where each empty symbol  $\blacksquare$  is replaced with an appropriate graphical placeholder.

*Note 2.* If  $\text{DAG}(D)$  is a tree, the resulting structure is simply a Merkle Tree [21] with some additional data taken under the hash (salted hashing).

*Note 3.* Observe that any Recipient of a selectively disclosed  $(\text{DAG}(D'), \sigma, L)$  can further restrict the content of the document and present it to another party. To blind an additional node  $A$  it suffices to change  $T(A)$  to  $\blacksquare$  and replace  $(\text{ID}_A, T_{\text{salt}}(A), 1)$  with  $(\text{ID}_A, \text{Hash}(T^+(A)), 0)$  in  $L$ .

*Note 4.* Similarly, for any Recipient of selectively disclosed  $(\text{DAG}(D'), \sigma, L)$ , it is infeasible to alter the content of the unblinded fields, as any such change would modify the final value of  $h_{\text{root}}$  or a collision of  $\text{Hash}$  would be found. Therefore, the security arguments reduce to the security of the underlying signature scheme.

*Note 5.* According to the Architecture and Reference Framework (ARF) [10], EDIW must support two mandatory standards for the electronic attestation of attributes with selective disclosure, namely ISO/IEC 18013-5 [18], which is generalized in ISO/IEC 23220-2 [19], and SD-JWT-based Verifiable Credentials [6]. In addition, EDIW may optionally support the W3C Verifiable Credentials Data Model v2.0 [27]. All three standards achieve selective disclosure by hashing each attribute together with a random value (once again, salted hashing). Thus, we are reusing the same concept for selective disclosure of signed documents as ARF uses for selective disclosure of electronic attestations of attributes.

### 3.3 Structured Encryption Approach

In this approach, all node labels of  $\text{DAG}(D)$  are first encrypted using an OTP-like symmetric encryption scheme<sup>8</sup> before being signed with **Sign**, while the keys used to encrypt the contents of the individual nodes of  $\text{DAG}(D)$  are distinct.

*Key derivation.* Assume that  $\text{DAG}(D)$  is a tree where the order of children of each node is defined.<sup>9</sup> The encryption keys are derived in a top-down manner:

1. Select a root key  $K_{\text{root}}$  at random, assign it to the root of  $\text{DAG}(D)$ ,
2. Let  $K_A$  be a key for node  $A$ , then  $m$  children of  $A$  are assigned, respectively, the keys
 
$$\text{Hash}(K_A, 1), \text{Hash}(K_A, 2), \dots, \text{Hash}(K_A, m). \quad (1)$$

*Encryption of  $\text{DAG}(D)$ .* We construct  $\text{EDAG}(D)$ , the encrypted copy of  $\text{DAG}(D)$ , by transforming the node labels according to the following procedure:

1. For each node  $A$  in  $\text{DAG}(D)$ , replace the label  $T(A)$  with

$$T'(A) := (T(A) \oplus K'_A, \text{Hash}(K_A, 0)),$$

where

- $K'_A$  is the output of a cryptographic PRNG seeded with  $K_A$ , truncated to the bit-length of  $T(A)$ ,
  - $\oplus$  denotes the bitwise XOR operation, and
  - $\text{Hash}(K_A, 0)$  is used for authenticating the key  $K_A$ .
2. Output  $(\text{EDAG}(D), K_{\text{root}})$ .

*Computing SHash.* In this case,  $\text{SHash}(D) := \text{Hash}(\text{ser}(\text{EDAG}(D)))$ , where **ser** can be any standard DAG serialization method.

*Signature creation.* A standard signature scheme **Sign** is applied, producing the signature  $\sigma$  with  $h := \text{SHash}(D)$ .

*Selective disclosure.* If the Signer wants to disclose the entire document  $D$  with its signature, then the Recipient gets the tuple

$$(\text{EDAG}(D), \sigma, K_0).$$

The Verifier can use  $K_0$  to derive all keys  $K_A$ , decrypt each  $T'(A)$  and finally check that the decrypted labels  $T(A)$  represent  $\text{DAG}(D)$ . The standard test is applied to  $\sigma$  and  $h$ .

Alternatively, the Signer can disclose only a set of nodes of  $\text{DAG}(D)$ , so that if a node  $A$  is disclosed, then automatically all its successors in  $\text{DAG}(D)$  will be disclosed as well. The following steps are executed:

<sup>8</sup> Note that any symmetric encryption scheme ( $\text{Enc}_K, \text{Dec}_K$ ) can be used in this scenario, additionally, if an AEAD scheme is used, appending authentication data can be neglected.

<sup>9</sup> We follow the idea from [32] of segment-based document protection.



1. Identify the nodes  $A_1, \dots, A_t$  such that the disclosed document  $D'$  contains  $T(A_1), \dots, T(A_t)$ . Let  $\mathcal{KS}$  denote the set of nodes  $A_1, \dots, A_t$  and all their successors.
2. Derive the keys  $K_{A_1}, \dots, K_{A_t}$  from  $K_0$  according to the original procedure.
3. Output the tuple

$$(\text{EDAG}(D), \sigma, \text{ID}_{A_1}, \dots, \text{ID}_{A_t}, K_{A_1}, \dots, K_{A_t}). \quad (2)$$

*Signature verification.* Let us focus on selectively disclosed  $D$ , as the case without selective disclosure corresponds to the case of choosing  $A_1$  being the root of  $\text{DAG}(D)$ .

For verification of a signature (2), the Recipient executes the following steps:

1. Verify the signature  $\sigma$  on  $\text{EDAG}(D)$ .
2. Starting from the nodes  $A_1, \dots, A_t$  and the keys  $K_{A_1}, \dots, K_{A_t}$ , calculate the key  $K_B$  for every successor node  $B$  contained in  $\mathcal{KS}$  using the procedure from Equation (1).
3. For  $A \in \mathcal{KS}$ , check the correctness of  $K_A$  by testing  $H(A) \stackrel{?}{=} \text{Hash}(K_A, 0)$ .
4. For  $A \in \mathcal{KS}$ , recover the plaintext  $T(A)$  by XOR-ing  $T(A) \otimes K'_A$  with  $K'_A$  derived from  $K_A$ . If a node  $B$  of  $\text{DAG}(D)$  does not belong to  $\mathcal{KS}$ , then the field  $T(B)$  should be replaced by the empty symbol  $\blacksquare$ .

*Note 6.* The visualization of  $\text{DAG}(D')$  in the Structured Encryption approach follows directly from the Merkle Tree approach. Also, observe that, similarly to the Merkle Tree approach, selective disclosure can be delegated. Namely, the user can derive certain descendant keys of  $K_{A_1}, \dots, K_{A_t}$  and use them for selective disclosure.

*Note 7.* It is infeasible for any Recipient of disclosed  $\text{EDAG}(D')$  to change the content of the fields, since to change the value  $T(A)$  it would be necessary to change  $K_A$ . However, this requires to find  $K'_A$  such that  $\text{Hash}(K'_A, 0) = \text{Hash}(K_A, 0)$  – otherwise  $\text{SHash}$  would be applied to a different string. All in all, a manipulation of the signed text requires finding a collision for  $\text{Hash}$ .

## 4 Future Work

Several important directions remain for future exploration. A proof-of-concept implementation of the proposed methods is needed to enable experimental evaluation of performance and scalability. Such a prototype should align with the ETSI standards [14] to facilitate adoption for legal transactions.

Furthermore, robust techniques for converting documents into DAGs require more attention and the creation of de facto standards. In this work, we only scratched the surface of representing text documents as DAGs. While translation from XML to DAG is relatively straightforward, handling arbitrary text formats (e.g., PDF or natural language text) is more challenging. Leveraging LLMs combined with traditional NLP techniques should also be investigated,

with careful consideration of factors such as reliability, privacy implications, and the inherent limitations of LLMs. Developing reliable methods for this translation could provide a powerful building block that could be used for multiple purposes, not only for signing.

Beyond the core framework, additional research should explore potential applications in knowledge extraction. One notable use case is proof of entitlement in whistleblowing scenarios [13]: a whistleblower has the right to submit a report if they are in any “work relation” with the reported organization. Such a relation may follow from various legal documents (employment contract, civil contract for services, confirmation of volunteer status, etc.). Presenting such a document in the original form during report submission can effectively thwart any pseudonymization attempt. This challenging problem has been recognized but so far not addressed in [13].

## 5 Final Remarks and Conclusions

This work demonstrates the feasibility of high-grain selective disclosure of signed documents derived from black-box signature schemes using a mapping from text documents to DAG structures.

To summarize, we suggest rethinking the concept of electronic signatures to enable selective disclosure of the data they contain without requiring re-signing. We have shown that such signatures can be implemented by slightly adapting existing cryptographic schemes, namely, incorporating standard (and in particular standardized) signature schemes, rather than designing entirely new ones. Such an approach reduces the standardization effort and facilitates faster adoption in practice.

The proposed method may be particularly useful for leveraging signed documents in terms of electronic attribute attestations, as introduced by eIDAS 2.0, in day-to-day administrative workflows following strictly the data minimization principle from GDPR. In our opinion, reshaping the concept of electronic signatures towards selective disclosure is a necessary condition to enable serious realization of GDPR.

## Acknowledgements

This research was partially funded by the National Science Centre, Poland under the OPUS call in the Weave programme [2023/51/I/ST6/02770]. For the purpose of Open Access, the author has applied a CC-BY public copyright licence to any Author Accepted Manuscript (AAM) version arising from this submission.

## References

1. Anthropic: Claude 3.7 Sonnet: Advanced Conversational AI Model. <https://www.anthropic.com/claude> (2025), Accessed: 2025-06-16

2. Bellare, M., Rogaway, P.: Random oracles are practical: a paradigm for designing efficient protocols. In: Proceedings of the 1st ACM Conference on Computer and Communications Security. pp. 62–73. CCS '93, Association for Computing Machinery, New York, NY, USA (1993). <https://doi.org/10.1145/168588.168596>, <https://doi.org/10.1145/168588.168596>
3. Bertoni, G., Daemen, J., Peeters, M., Assche, G.V.: Cryptographic Sponge Functions. Technical Report, Version 0.1 CSF-0.1, STMicroelectronics and NXP Semiconductors, — (jan 2011), <https://keccak.team/files/CSF-0.1.pdf>, version 0.1 (Jan. 14, 2011)
4. Beuchat, J.L., Rexhepi, V.: A Digital Identity in the Hands of Swiss Citizens. Cryptology ePrint Archive, Paper 2023/1099 (2023), <https://eprint.iacr.org/2023/1099>
5. Boneh, D., Boyen, X.: Short Signatures Without Random Oracles. Cryptology ePrint Archive, Paper 2004/171 (2004), <https://eprint.iacr.org/2004/171>
6. Campbell, B., Fett, D., Terbu, O.: SD-JWT-based Verifiable Credentials (SD-JWT VC). Internet-Draft draft-ietf-oauth-sd-jwt-vc-09, IETF (2025), <https://datatracker.ietf.org/doc/draft-ietf-oauth-sd-jwt-vc/>, work in progress, expires 28 November 2025
7. Chaum, D.: Security without identification: transaction systems to make big brother obsolete. Communications of the ACM **28**(10), 1030–1044 (Oct 1985). <https://doi.org/10.1145/4372.4373>, <http://dx.doi.org/10.1145/4372.4373>
8. DeepSeek: DeepSeek R1: Open-Source Reasoning Model. <https://www.deepseek.com> (2025), Accessed: 2025-06-16
9. ETSI: Electronic Signatures and Infrastructures (ESI); Cryptographic Suites. ETSI Technical Specification ETSI TS 119 312 V1.4.3, ETSI (Aug 2023), [https://www.etsi.org/deliver/etsi\\_ts/119300\\_119399/119312/01.04.03\\_60/ts\\_119312v010403p.pdf](https://www.etsi.org/deliver/etsi_ts/119300_119399/119312/01.04.03_60/ts_119312v010403p.pdf), version 1.4.3 (2023-08)
10. EU Digital Identity Wallet Community: Architecture and Reference Framework for the European Digital Identity Wallet. <https://github.com/eu-digital-identity-wallet/eudi-doc-architecture-and-reference-framework> (jun 2025), version 2.1.0, released June 6, 2025
11. European Commission: Commission Implementing Decision (EU) 2015/1506. Official Journal of the European Union (sep 2015), [https://eur-lex.europa.eu/eli/dec\\_impl/2015/1506](https://eur-lex.europa.eu/eli/dec_impl/2015/1506)
12. European Commission: Regulation (EU) 910/2014. <https://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:52021PC0281> (2021)
13. European Parliament and Council of the European Union: Directive (EU) 2019/1937 (2024), <https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX%3A32019L1937>, Accessed: 2024-05-27
14. European Telecommunications Standards Institute: Electronic Signatures and Infrastructures (ESI); Certificate Profiles; Part 1: QCProfile. ETSI Standard EN 319 162-1 V1.1.1, ETSI (2016), [https://www.etsi.org/deliver/etsi\\_en/319100\\_319199/31916201/01.01.01\\_60/en\\_31916201v010101p.pdf](https://www.etsi.org/deliver/etsi_en/319100_319199/31916201/01.01.01_60/en_31916201v010101p.pdf), Accessed: 2025-06-27
15. Holt, J.E., Seamons, K.E.: Selective disclosure credential sets. Cryptology ePrint Archive, Paper 2002/151 (2002), <https://eprint.iacr.org/2002/151>
16. International Civil Aviation Organisation: Machine Readable Travel Documents - Part 11: Security Mechanism for MRTDs. Doc 9303 (2021)
17. International Organization for Standardization, International Electrotechnical Commission: IT Security techniques – Digital signatures with appendix – Part

- 3: Discrete logarithm based mechanisms (ISO/IEC 14888-3:2018) (– 2018), <https://www.iso.org/standard/76382.html>
18. International Organization for Standardization, International Electrotechnical Commission: ISO/IEC 18013-5:2021 Personal identification – ISO-compliant driving licence – Part 5: Mobile driving licence (mDL) application (2021), <https://www.iso.org/standard/69084.html>, published 18 August 2021; interface specs for mobile driving licence
19. International Organization for Standardization, International Electrotechnical Commission: ISO/IEC TS 23220-2:2024 (2024), <https://www.iso.org/standard/86782.html>, technical specification; first edition published November 2024
20. Lamport, L.: Constructing Digital Signatures from a One Way Function (2016), <https://api.semanticscholar.org/CorpusID:59679804>
21. Merkle, R.C.: A Certified Digital Signature. In: Advances in Cryptology — CRYPTO '89 Proceedings. Lecture Notes in Computer Science, vol. 435, pp. 218–238. Springer (1989). [https://doi.org/10.1007/0-387-34805-0\\_24](https://doi.org/10.1007/0-387-34805-0_24)
22. Moriarty, K., Kaliski, B., Jonsson, J., Rusch, A.: PKCS #1: RSA Cryptography Specifications Version 2.2. Internet Engineering Task Force (IETF) (2016)
23. National Institute of Standards and Technology: Digital Signature Standard (DSS), FIPS Publication 186-5. Federal Information Processing Standard 186-5, Gaithersburg, MD (feb 2023). <https://doi.org/10.6028/NIST.FIPS.186-5>, <https://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-5.pdf>
24. OpenAI: GPT-4o: A Multimodal Large Language Model. <https://openai.com/chatgpt> (2024), Accessed: 2025-06-16
25. Python Software Foundation: xml.etree.ElementTree — The ElementTree XML API (2024), <https://docs.python.org/3/library/xml.etree.elementtree.html>, python 3.12 documentation
26. Rankl, W., Effing, W.: Handbuch der Chipkarten - Aufbau, Funktionsweise, Einsatz von Smart Cards (4. Aufl.). Hanser (2002)
27. Sporny, M., Longley, D., Chadwick, D., Steele, O., Sabadello, M., Reed, D.: Verifiable Credentials Data Model v2.0. W3C Recommendation W3C TR VC-DATA-MODEL-2.0, W3C Verifiable Credentials Working Group (may 2025), <https://www.w3.org/TR/vc-data-model-2.0/>, version of 15 May 2025
28. Tananaev, A.: passport-reader: e-Passport NFC Reader Android app. <https://github.com/tananaev/passport-reader> (2023), <https://github.com/tananaev/passport-reader>, version 3.1
29. The European Parliament and the Council of the European Union: Regulation (EU) 910/2014. Official Journal of the European Union **257/73** (2014), [https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3A0J.L\\_.2014.257.01.0073.01.ENG](https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=uriserv%3A0J.L_.2014.257.01.0073.01.ENG)
30. The European Parliament and the Council of the European Union: Regulation (EU) 2016/679. Official Journal of the European Union **119**(1) (2016), <http://eur-lex.europa.eu/legal-content/EN/TXT/HTML/?uri=CELEX:32016R0679&from=EN#d1e3265-1-1>
31. Wrinkl, W., Effing, W.: Smart Card Handbook (2004), <http://deadnet.se:8080/Books%20and%20Docs%20on%20Hacking/Electronics/Smart%20Card%20Handbook.pdf>, third Edition
32. Xu, D., Tang, Z., Yu, Y.: An efficient key management scheme for segment-based document protection. In: 2011 IEEE Consumer Communications and Networking Conference, CCNC 2011, Las Vegas, NV, 9-12 January, 2011. pp. 896–900. IEEE (2011). <https://doi.org/10.1109/CCNC.2011.5766636>, <https://doi.org/10.1109/CCNC.2011.5766636>

# Invisible Encryption

Shahzad Ahmad<sup>1</sup>[0000-0002-9654-869X], Stefan Rass<sup>1,2</sup>[0000-0003-2821-2489], and Zahra Seyedi<sup>3</sup>[0009-0002-8492-4640]

<sup>1</sup> LIT Secure and Correct Systems Lab, Johannes Kepler University, Linz, Austria [shahzad.ahmad@jku.at](mailto:shahzad.ahmad@jku.at)

<sup>2</sup> Institute for AI and Cybersecurity, University of Klagenfurt, Klagenfurt, Austria [stefan.rass@jku.at](mailto:stefan.rass@jku.at)

<sup>3</sup> Department of Electronics, Information and Bioengineering, Polytechnic University of Milan, Milan, Italy  
[zahrasadat.seyedi@mail.polimi.it](mailto:zahrasadat.seyedi@mail.polimi.it)

**Abstract.** We present Invisible Encryption, a cryptographic protocol that camouflages the sharing of a secret within standard encrypted traffic to avoid detection in monitored environments. This paper introduces Invisible Encryption, a novel protocol integrating threshold secret sharing, steganography, and public-key cryptography to enable covert communication. By embedding a secret share within standard encrypted traffic, specifically by disguising it as a session key or nonce in a hybrid encryption scheme, our method ensures that the transmission of the secret remains undetectable. The secret is reconstructed from shares derived from a public natural language text and the transmitted share, with the selection of shares protected by a secret seed. We provide a formal security analysis, demonstrating that Invisible Encryption achieves confidentiality and plausible deniability under standard cryptographic assumptions. Invisible Encryption offers a robust solution for applications that require secure, undetectable communication, such as censorship-resistant systems and whistleblower protection.

**Keywords:** Secret sharing · Steganography · Covert communication · Plausible deniability.

## 1 Introduction

In environments where the very existence of encrypted communication raises suspicion or invites censorship, it is crucial to conceal both the content *and* the purpose of messages. Traditional steganography hides data within innocuous media (images, audio, etc.), but often requires specialised embedding methods and can be detected via statistical analysis. In contrast, Invisible Encryption repurposes standard cryptographic envelopes and protocols as carriers of hidden information; in other words, can we use cryptography itself as a steganographic channel?

A different way to frame the challenge is: while ciphertexts are often easy to recognise as such (e.g., random-looking strings in a log file), can we construct an encryption function that maps a meaningful natural-language plaintext into a ciphertext that itself appears to be a natural-language text? For example, could even this paragraph contain an encrypted message by means other than classical steganography? The answer is not straightforward. Even the simplest substitution cipher (e.g., replacing words of a sentence with other words from the same or another language) demonstrates the difficulty: even if the “components” of the ciphertext are valid words, their combination, as dictated by grammar, will almost surely form nonsense rather than a meaningful sentence. Large language models seem promising to generate fluent text, but their design is not intended for cryptographic reversibility.

However, suppose we allow the target “ciphertext” to exist **before** encryption, i.e.. In that case, we choose a cover text in advance, and we can design our encryption to map a given plaintext to a target sequence of values that forms that meaningful text. It turns out that **polynomial secret sharing** offers a way to do this: we treat words from a natural-language text as shares *values* on a secret-sharing polynomial, and we let the arguments (indices) at which those shares are defined be determined pseudorandomly by a secret seed. We also include an additional single share to ensure the plaintext message can be recovered. To see why this extra share is needed, suppose we naively tried to reconstruct a secret message  $m$  by XOR’ing together words from the cover text:  $m = w_1 \oplus w_2 \oplus \dots \oplus w_n$ . Unless those words were specially chosen (and used only once), this is unlikely to equal  $m$  except by coincidence. Using repeated plaintext words as a one-time pad invites the classic Friedman attack. Instead, we introduce an additional random component  $r$  such that  $m = w_1 \oplus \dots \oplus w_n \oplus r$ . This  $r$  will appear random and not part of any natural language.

Our strategy is to *disguise* this extra random share  $r$  as a legitimate random value in a standard cryptographic protocol transcript. Random nonces, session keys, or other random outputs are common in regular encrypted traffic and typically do not arouse suspicion. For example, imagine Alice and Bob exchange mostly plaintext emails but occasionally perform an authenticated key exchange or send an encrypted attachment. What if we let the words  $w_1, \dots, w_n$  be drawn from the plaintext email conversation, and let the final random value  $r$  be transmitted as part of the cryptographic protocol (say, as a session key in a hybrid encryption)? In other words, even though  $r$  is a share in an  $(n+1)$ -out-of- $(n+1)$  secret sharing scheme, its transmission is camouflaged as a random cryptographic artefact. A simple choice is to piggyback on a hybrid encryption or challenge-response authentication protocol – for instance, use  $r$  as the “session key” in an RSA-based key exchange.

The key insight is that transmissions of random bits are routine in cryptographic protocols and thus unlikely to be flagged as suspicious, whereas sending plaintext secret data would betray that something hidden is being communicated. More importantly, if an eavesdropper intercepts the entire communication, they will observe normal-looking plaintext messages (the cover text) and a standard cryptographic exchange (e.g., an RSA-encrypted session key and a ciphertext of a decoy message). This should not trigger special scrutiny: the adversary sees nothing beyond an ordinary encrypted session amid otherwise plaintext conversation. Even if the adversary captures all the traffic, a single random session key is insufficient to recover any secret message.

In summary, the Invisible Encryption protocol enables covert communication by concealing a secret within standard encryption traffic. The hidden secret share is indistinguishable from the random values usually present in cryptographic protocols, providing **plausible deniability**: participants can claim the exchange was purely routine (e.g., an encrypted email or authentication step). An adversary who is unaware of the secret seed cannot identify which parts of the cover text are carrying hidden information or reconstruct the secret. The primary contributions of this work are as follows:

1. **A new form of steganography that uses cryptography as its carrier**: Integrates Shamir’s  $(k, n)$ -threshold secret sharing with public-key and symmetric encryption, embedding a secret share within a decoy message to enable reconstruction from a public cover text.
2. **Plausible Deniability**: Permits participants to claim the communication involves routine encryption, safeguarding against adversarial or legal scrutiny.

3. **Efficient Implementation:** Provides a Python prototype with execution times under 400 ms and a communication overhead of 544 bytes, suitable for real-time, bandwidth-constrained applications <sup>4</sup>.

These contributions enable secure, undetectable communication, applicable to censorship-resistant systems, whistleblower protection, and privacy-preserving environments.

The paper is organized as follows: Section 2 surveys relevant work in secret sharing, steganography, and covert communication. Section 3 outlines the mathematical and cryptographic preliminaries used in our scheme. Section 4 describes the Invisible Encryption scheme in detail, including algorithms for setup, share derivation, encryption, and decryption. Section 5 defines the system and adversary model and formalizes the security goals. Section 6 presents the security analysis of the scheme. Section 7 discusses our prototype implementation, including the protocol flow and performance measurements. Finally, Section 8 concludes the paper and highlights directions for future research (such as extending the scheme to multiple messages under one key).

## 2 Related Work

The foundation of Invisible Encryption builds upon several decades of research in threshold secret sharing, tracing back to Shamir [32] and Blakley [7], which introduced polynomial interpolation and geometric approaches, respectively. Subsequent enhancements include verifiable secret sharing by Feldman [19] and Pedersen [29], proactive renewal of shares by Herzberg et al. [22], and computational secret sharing by Krawczyk [26]. More recently, Komargodski et al. [24] extended these ideas to threshold fully homomorphic encryption, paving the way for secure computation on shared data.

The linguistic obfuscation in Invisible Encryption builds on techniques from linguistic steganography. Early mimic functions, as demonstrated by Wayner [37] and lexical methods by Chapman and Davida [12], have shown how ciphertext can be disguised as natural text. Advances by Chang and Clark [11], and Safaka et al. [31] exploit syntactic and semantic transformations for watermarking and covert channels. Unlike prior work that embeds payloads by modifying cover text, Invisible Encryption maps existing text to shares through selective indexing and hashing, enhancing naturalness and deniability.

Threshold cryptography enables collaborative cryptographic operations without revealing private keys, as pioneered by Desmedt and Frankel [16] and refined by Shoup [34]. Practical frameworks, such as FROST by Komlo and Goldberg [25], demonstrate efficient threshold signatures. Invisible Encryption’s hybrid encryption follows the RSA-based KEM/DEM paradigm of Rivest et al. [30] and the formal models of Cramer and Shoup [14], with extensions from Abdalla et al. [2] and identity-based methods by Watanabe et al. [36]. Its use of encrypted decoy messages for plausible deniability draws on TrueCrypt’s hidden volumes [1], deniable encryption by Canetti et al. [10], and schemes by Tyagi et al. [35], and Dodis et al. [18].

Indistinguishability obfuscation, introduced by Barak et al. [4] and Garg et al. [20], as well as functional encryption by Boneh et al. [9], share the goal of hiding information while preserving utility. Its threshold structure echoes MPC protocols from Yao [38], Goldreich et al. [21], SPDZ by Damgård et al. [15], Sharemind by Bogdanov et al. [8], and integrating secret sharing in MPC by

<sup>4</sup> Here is the link for the Python proof-of-concept implementation: <https://github.com/shahzadssg/Invisible-Encryption.git>

Benaloh [5] and Cramer et al. [13]. Instances of invisible encryption using RSA-based carriers can be vulnerable to Shor’s algorithm [33], motivating post-quantum alternatives such as NTRU [23], Ring-LWE [27], McEliece [28], Rainbow [17], and threshold lattice schemes by Bendlin et al. [6]. Given the crucial threats induced by quantum computing, the long-term security of the protocol will depend on transitioning to post-quantum cryptography, a task that is noted here as essential future work but is not integrated into the current model.

### 3 Preliminaries

We let words from our natural language appear in some fixed binary encoding, e.g., ASCII code, Unicode or others, such that we can uniquely associate  $w \in \{0, 1\}^*$  with some meaningful string (at least a symbol, up to a whole natural language word). Furthermore, let  $|w|$  be the length of  $w$  in bits, so that treating  $w$  as an integer in binary notation, we have  $w \leq 2^{|w|}$ . Let us fix a prime  $p$  and length  $n \in \mathbb{N}$  such that all words  $w \in \{0, 1\}^n$  in our plaintexts<sup>5</sup>, treated as integers, fit into the range  $0 \leq w \leq p-1$ , so that our plaintext (natural language words) are canonically interpretable as elements of  $\mathbb{F}_p$  for otherwise, we may take a word’s hash value modulo  $p$  to map it into an element of  $\mathbb{F}_p$ ; as our practical implementation does.

We let a natural language text be given as an ordered sequence of  $L \in \mathbb{N}$  words  $T = (w[0], w[1], \dots, w[L]) \in (\{0, 1\}^*)^n$ , all possibly padded to the same (maximum) length. Within this sequence of words, we will embed our secret and, if necessary, extend the sequence with additional entries. Let  $t \in \mathbb{N}$  be a security parameter, and let the length of the carrier text  $T$  be  $n = \text{poly}(t)$  depend on  $t$  by some (fixed) polynomial  $\text{poly}$ . Furthermore, let the plaintext be a string of length  $m$ , where  $m = \text{poly}(t) < n(t)$ , that also depends on  $t$  by some (other) polynomial. Table 1 provides an overview of variables and functions appearing throughout the construction.

*Polynomial Secret Sharing:* A secret  $m \in \mathbb{F}_p$  is shared using a polynomial  $P(x) = m + a_1x + \dots + a_{k-1}x^{k-1}$ , with  $k$  distinct evaluations  $(x_i, P(x_i))$  sufficient for reconstruction. The Lagrange interpolation formula reconstructs  $P(0)$ :

$$m = P(0) \equiv \prod_{j=1}^k P(x_j) \cdot \prod_{i=1, i \neq j}^k \frac{-x_i}{x_j - x_i} \pmod{p}.$$

*Pseudorandom Number Generator (PRNG):* A PRNG is a deterministic algorithm  $G : \{0, 1\}^s \rightarrow \{0, 1\}^\ell$  that takes a seed of length  $s$  and produces a longer pseudorandom output of length  $\ell$ . A PRNG is secure if its output is computationally (in polynomial time in  $\ell$ ) indistinguishable from a truly random string of length  $\ell$ .

*(Non-)Cryptographic Disguise of random strings:* to unsuspiciously hide a random string in natural language text, such as within a log file or similar, we can combine asymmetric encryption (e.g., RSA-OAEP) with symmetric encryption (e.g., AES-CBC) following the KEM/DEM paradigm. The RSA component encrypts the seed and the freshly generated secret share  $s_{\text{new}}$ , as necessary by our introductory argument above; the AES component encrypts a decoy message using  $s_{\text{new}}$  as the key with a random  $IV$ . That use of hybrid encryption is arbitrary here; any cryptographic (or other)

<sup>5</sup> mildly assuming that the plaintext to carry our secret is from a natural language whose words will not have unbounded lengths



Table 1: List of symbols used in the construction.

Symbol	Description
$x  y$	concatenation of strings $x$ and $y$
$L$	Length of the cover text $T$ (number of words)
$w[i]$	$i$ -th word in the cover text $T$
$T$	Original cover text, an ordered sequence of words $(w[1], \dots, w[L])$
$T'$	Updated cover text (ciphertext), i.e., $T    w[L+1]$ after embedding the final share
$k$	Threshold for secret reconstruction
$m$	Secret message in $\mathbb{F}_p$ to be shared, resp. covertly transmitted
$P(x)$	Degree- $(k-1)$ polynomial used for secret sharing, satisfying $P(0) = m$
$x_j$	Pseudorandom abscissa values, generated via $x_j = H(x_{j-1})$
$x_{\text{new}}$	Fresh abscissa for the additional share, derived as the next hash in the chain
$s_j$	Share values $s_j = P(x_j)$ (for $1 \leq j \leq k-1$ ), or $s_{\text{new}} = P(x_{\text{new}})$
$s_{\text{new}}$	Additional share computed from interpolated polynomial
$H$	Cryptographic hash function modeled as a random oracle
$PRNG$	Pseudorandom number generator seeded with $x_0$ , used to pick distinct word indices
$x_0$	Secret seed used for pseudorandom generator $PRNG$
<b>Param</b>	Public parameters tuple $(p, H, PRNG)$
<b>SK</b>	Secret key, consisting of $(x_0, k)$ shared by sender and receiver

protocol  $\Pi$  that at some point transmits a random string between parties would be admissible. One example is challenge-response authentication, where Alice may ask Bob to decrypt (symmetrically) or digitally sign (asymmetrically) a random nonce, which is a share in the above polynomial sharing scheme. As a non-cryptographic example, that would even work inside a transmitted file only, e.g., letting the plaintext be in a PDF with embedded content; we can even include a QR code in a text that contains a weblink, inside which a random (e.g., base-64 encoded) login-token is embedded (although this may negatively affect the security if the key is re-used; see Section ??). The “login” token in the weblink can be the additional share to be transmitted, while the actual URL will open accordingly (or not) solely for the sake of deception. Suppose a login is attempted by clicking on the web link. In that case, this mechanism may even serve as an intrusion detection signal, as the receiver already knows that the QR code contains a share and should not be opened. In contrast, the adversary may not know this and may discover it upon this trial.

## 4 The Invisible Encryption Scheme

Figure 1 illustrates the basic idea: like in Shamir’s scheme [32], we let  $P(x) = m + a_1x + \dots + a_{k-1}x^{k-1}$  be a polynomial, in which we fix  $m \in \{0, 1\}^*$  to be the secret. However, different to the classical instance of Shamir’s sharing, we fix a series of values of the polynomial, rather than its coefficients. That is, we will look for a polynomial for which  $P(x_i) = w[i]$ , for some value  $x_i$  and some word  $w[i]$  from the natural language text, for a total of  $\geq k$  but  $\leq n$  values  $x_i$  and words  $w[i]$ .

While it would be conceptually trivial to fix a sequence  $x_1, \dots, x_{k-1}$  and words  $w[i_1], \dots, w[i_{k-1}]$  from the text and interpolate a polynomial by solving a linear system of equations, the reconstruction would then require the same set of values again. This knowledge would have to be transmitted secretly from Alice to Bob, but (i) could not be used a second time, and (ii) would be more data than transmitting the secret directly. To overcome both issues, we let the sequence of  $x_1, \dots, x_{k-1}$

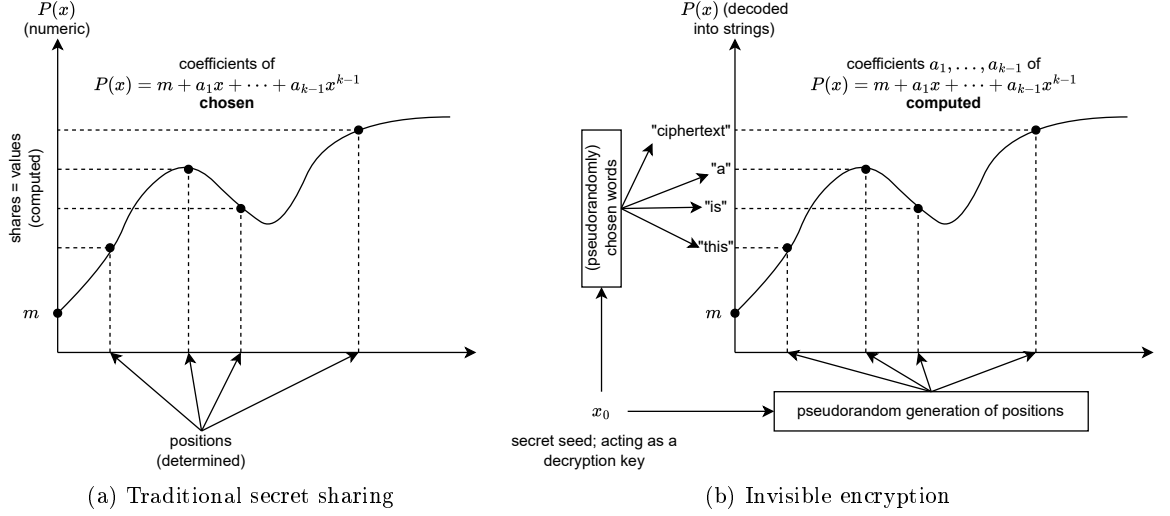


Fig. 1: Conventional secret sharing, vs. invisible encryption: while in the conventional setting, the polynomial is chosen, and shares are its values taken at (defined or secret) positions, invisible encryption twists around this process by choosing the natural language words to be the values that the polynomial shall take, and interpolating it at positions that are defined secretly from a seed value  $x_0$ . The seed can also determine the words to interleave the shares with (the cover text to embed the ciphertext), and the polynomial is then finally determined by interpolation. Further shares are then computed in the traditional form again (and are required for proving the steganographic security).

and  $i_1, \dots, i_{k-1}$  be generated pseudorandomly based on a (fixed-size) secret  $x_0$  that will act as a secret decryption key. The computation of the polynomial remains a direct interpolation, but to recover  $m$  as the plaintext secret will require at least one additional share (as explained in the introductory intuition).

#### 4.1 Formal Scheme Definition

We model Invisible Encryption as a triple of probabilistic algorithms: (**Setup**, **Enc**, **Dec**), defined as follows:

**Setup**( $t, L$ ): On input a security parameter  $t \in \mathbb{N}$  and the cover-text length  $L = \text{poly}(t)$ , it selects:

- i) A large prime  $p$  of bit-length  $\lambda = \text{poly}(t)$  and finite field  $\mathbb{F}_p$ .
- ii) A hash function  $H : \{0, 1\}^* \rightarrow \mathbb{F}_p$ .
- iii) A cryptographically secure pseudorandom number generator  $PRNG$ , whose output range is  $\{1, \dots, L\}$ .
- iv) A secret seed  $x_0 \in \{0, 1\}^\lambda$  uniformly at random (denoted  $x_0 \xleftarrow{\$} \{0, 1\}^\lambda$ ).
- v) A threshold  $k \xleftarrow{\$} \{2, \dots, L\}$ . The symbol  $x \xleftarrow{\$} X$  denote a random uniform selection of  $x \in X$ .
- vi) Parameters  $\text{Param} = (p, H, PRNG)$ .

**Output** ( $\text{Param}, \text{SK} = (x_0, k)$ ).

We remark that the hash function  $H$  and the pseudorandom number generator  $PRNG$  serve the same purpose, but deliver different “items”, i.e., either the points on the abscissa (by a hash chain) or the selection of words from the text (via the  $PRNG$ ). We distinguish the two for the sole purpose of indicating the meaning of their outputs directly by the symbol (easier than distinguishing two hash functions or two PRNGs).

$\text{Enc}(\text{PK}, m, T)$ : On input public parameters  $\text{Param}$ , secret  $m \in \mathbb{F}_p$ , and cover text  $T = (w[1], \dots, w[L])$  of length  $L$ , it:

- i) Derives  $n$  shares  $\{(x_j, s_j)\}_{j=1}^{k-1}$  from  $(x_0, T)$  by:
  - a) pseudorandomly computing the values  $x_j \leftarrow H(x_{j-1})$  for  $j = 1, 2, \dots$  until  $k-1$  distinct indices are found (skipping duplicate values, if any).
  - b) pseudorandomly picking  $k-1$  distinct indices  $i_j \in \{1, \dots, L\}$  using the PRNG seeded with  $x_0$  and setting  $s_j \leftarrow w[i_j]$  for  $j = 1, 2, \dots, k-1$  (duplicates are allowed here).
- ii) Interpolates the unique polynomial  $P \in \mathbb{F}_p[x]$  of degree  $< k$  satisfying

$$P(0) = m, \quad P(x_j) = s_j \quad \text{for the } k-1 \text{ points } (x_j, s_j) \text{ chosen in step i)}$$

- iii) Computes a fresh abscissa  $x_{\text{new}} = H^k(x_0)$  (continuation of the hash chain; if  $x_{\text{new}} \in \{0, x_1, \dots, x_{j-1}\}$  from above, then we iterate further until we get an “unused” nonzero value), and put  $s_{\text{new}} \leftarrow P(x_{\text{new}})$ .
- iv) Execute the protocol or other procedure  $\Pi$  that will use  $s_{\text{new}}$  (e.g., as a session key, during an authentication, embedded in a QR code, etc.), yielding  $w[L+1] \leftarrow \Pi(M)$  as the transcript of the protocol  $\Pi$ , which is now added (e.g., appended) to the existing cover text<sup>6</sup>, giving  $T' \leftarrow T \| w[L+1] = (w[1], \dots, w[L], w[L+1])$ .

**Outputs** the updated cover text  $T'$  as the ciphertext, decryptable under the secret key  $\text{SK} = (x_0, k)$ .

$\text{Dec}(\text{SK}, C, T')$ : On input of  $\text{SK} = (x_0, k)$ , and cover-text  $T' = (w[1], \dots, w[L], w[L+1])$ , we do the following:

- i) Extract  $s_{\text{new}}$  from  $w[L+1]$ , and recover all points  $\{(x_i, w[i])\}_{i=1}^k$  by recomputing the pseudorandom sequences just as done during the encryption.
- ii) Interpolate  $P$  through the just created points, and return the secret message  $m \leftarrow P(0)$

The encryption of multiple messages under the (same) secret key  $\text{SK} = (x_0, k)$  works likewise but will produce only  $s_{\text{new}}$  afresh for any further messages, upon using a distinct polynomial  $P$  for each message. The remaining set of shares will be the same (due to the same seed to determine the pseudorandom sequences). We will revisit this issue in Section ??.

## 5 System Model and Threat Assumptions

Building on the algorithms and syntax defined in Section 4, we now present a formal security model. We specify the roles, the information each party controls, the adversary’s oracle access and resource bounds, the information flow, and the precise security objectives.

<sup>6</sup> in many practical cases,  $\Pi$  may have a much longer transcript than what would fit into a string of the “block” size  $n$  that we fixed; in that case, we may add further blocks accordingly, but w.l.o.g., let the transcript (the relevant part of it), appear as  $w[L+1]$  to simplify the notation

### 5.1 Adversary Model and Security

Let  $\mathcal{A}$  be a probabilistic polynomial-time (PPT) adversary who knows all public parameters and the complete ciphertext  $T' = (w[1], \dots, w[L], w[L+1])$ , including all details about the procedure (cryptographic protocol or other)  $\Pi$ , whose transcript appears in  $T'$ .

The adversary's goal is to recover the secret  $m \in \mathbb{F}_p$ . The security of the linguistic obfuscation component depends on the indistinguishability of the mapped shares from random elements in the text. Computational indistinguishability is understood in the usual cryptographic sense, i.e., every polynomial-time-bounded probabilistic attacker would have a negligible chance of discovering the secret if the security parameter  $t$  becomes sufficiently large. We call a function  $\nu : \mathbb{N} \rightarrow \mathbb{N}$  negligible if it decays faster than any polynomial, i.e., if for every  $c > 0$ , there is some  $n_c \in \mathbb{N}$  such that  $\nu(n) \leq n^{-c}$  for all  $n \geq n_c$ .

**Definition 1 (Steganographic Security).** *A steganographic system is secure if the distribution of cover objects (text) and stego objects (text with embedded information) are computationally indistinguishable. More formally, let  $t \in \mathbb{N}$  be a security parameter, and let  $p, q$  be non-constant strictly positive polynomials in  $t$ . Given a carrier text  $T \in (\{0,1\}^*)^{p(t)}$  and whose total length is  $\leq q(t)$ , which embeds a secret  $x \in \{0,1\}^n$  of fixed size  $n \leq p(t)$ , we call it's embedded secret steganographically secure if  $\Pr(\text{adversary correctly outputs } x \mid \text{given } T) \leq \nu(t)$ , where  $\nu(t)$  is negligible for a polynomially (in  $t$ ) time-bounded attacker.*

## 6 Security Analysis

The security of our scheme will hold in a computational, not information-theoretic, sense under a random oracle assumption on the hash function  $H$  and the pseudorandom generator  $PRNG$ .

### 6.1 Security of the Secret Sharing Scheme (is only computational)

Shamir's Secret Sharing provides perfect information-theoretic security, meaning that even an adversary with unlimited computational power cannot determine the secret from fewer than  $k$  shares. However, our construction cannot retain this property because the plaintext words will not exhaust the entirety of elements in  $\mathbb{F}_p$ , and (more importantly), the adversary, unlike in the traditional setting of secret sharing, is in possession of all shares; only does not know which are the right ones. Hence, the usual argument for information-theoretic security that for every possible secret, a missing share would exist to produce exactly this secret will fail since all possibilities are already "fixed." However, we do retain an intractably large search space for the adversary since every subset of (ciphertext) elements could be the set of desired shares, and their entire count is  $O(2^L)$  if the threshold is unknown. Letting the cover text be reasonably long, this will eventually exceed a polynomial bounded attacker's capabilities. If the number  $k$  of shares would be known to the adversary, then the search space "shrinks" to a size of  $O(n^k)$  (containing all  $k$ -element subsets, but fewer, since the "cryptographic protocols transcript" must be part of the sharing, reducing the number of shares by 1, or a relatively smaller choice of possibilities, at least).

### 6.2 Steganographic Security of Invisible Encryption

The text-to-shares mapping in Invisible Encryption is steganographically secure under the random oracle model, specifically:

**Theorem 1.** *The scheme from Section 4 is steganographically secure against a polynomially time-bounded attacker for the encryption of a single message, provided that:*

- (random oracle assumption)  $H$  and  $PRNG$  behave as random oracles
- (known ciphertext assumption) The adversary does not know  $SK = (x_0, k)$ , but knows the entire cover text  $T' = (w[1], \dots, w[L], w[L+1])$  and details about  $\Pi$ , whose transcript appears as  $w[L+1]$  (possibly more, if the transcript is longer than one “word” in the cover text).

*Proof.* The attacker can correctly recover the secret if and only if two conditions are met:

- (A) correctly guesses the right words from the text,
- (B) correctly gets the sequence  $x_1, \dots, x_{k-1}, x_{\text{new}}$

Since the number  $k \in \{2, \dots, L\}$  is unknown to the attacker, which, by a random oracle assumption on  $PRNG$ , leaves it with a uniform choice of any subset of  $\{1, \dots, L\}$  to be the candidate set of shares. This number of  $O(2^L)$  many choices makes the search space super-polynomially large since we also assumed  $L = \text{poly}(t)$  (see Section 4.1). The chances to guess the correct subset thus become  $\Pr(A) = 2^{-L} = 2^{-\text{poly}(t)}$ .

Since this sequence is as well pseudorandom, its seed  $x_0$  is unknown, and  $H$  acts like a random oracle, the chances to recover the sequence are the same as for guessing  $x_0$ , making the probability of accomplishing the second condition  $\Pr(B) = 2^{-\lambda} = 2^{-\text{poly}(t)}$ . Thus, the overall chances for the attacker to recover  $m$  correctly come to  $\Pr(A \wedge B) \leq \min\{\Pr(A), \Pr(B)\} = \min\{2^{-\text{poly}(t)}, 2^{-\text{poly}(t)}\} \leq 2^{-\text{poly}(t)}$ , which is negligible (where  $\text{poly}$  can denote three distinct polynomials in the last expression).

### 6.3 Plausible deniability

Immanent to the design of the scheme is the ability to plausibly deny the hidden message inside it, since the cover text goes unmodified, and any data embedded in pictures, cryptographic protocols or other parts (that allow unsuspiciously transmitting bit strings of any form) will contain only a share that contains no information (as it is information-theoretically insufficient to recover the secret). If strong coercion is anticipated, embedding multiple secrets for decoy or deniability purposes is another option; this method has previously been described in [3], with proven security and deniability.

## 7 Example Implementation

We developed a proof-of-concept implementation of Invisible Encryption in Python to validate its correctness and evaluate performance. In this section, we describe the implementation details, illustrate the protocol flow between sender and receiver (with a diagram), and present an analysis of covertness and runtime performance.

### 7.1 Environment and Tools

The prototype was implemented in Python 3.8 and tested on a standard workstation (2.4 GHz Intel Core i5 CPU, 16 GB RAM). We used the Galois library for finite field arithmetic and the Python cryptography library for RSA and AES operations. The cover texts in our tests were drawn from sample English texts (e.g., Wikipedia articles) to simulate natural communications.

## 7.2 Implementation Details

The prototype implements the algorithms outlined in Section 4, modularised for clarity and reusability. The core components, reflecting the steps described in the formal definitions, are detailed below, along with the specific libraries and parameters used in our Python implementation.

*Setup and Key Generation:* This phase initialises the necessary parameters and keys.

- i) **Field and Hash Initialization:** A large prime  $p$  is sampled to define the finite field  $\mathbb{F}_p$ . In our implementation, a 256-bit prime  $p$  was generated using the `galois` library to ensure a sufficiently large field for mapping potential word hashes. The hash function  $H : \{0, 1\}^* \rightarrow F_p$  is defined as the SHA-256 hash of the input mapped into the field by taking the result modulo  $p$ . Specifically,  $H(m) = \text{SHA-256}(m) \text{ MOD } p$ .
- ii) **Secret Seed and Threshold Selection:** A 256-bit secret seed  $x_0$  is generated using `os.urandom(32)`. For selecting word indices from the cover text, our proof-of-concept uses Python’s built-in `random` module, seeded with  $x_0$ . It is important to acknowledge that this standard PRNG is not cryptographically secure and does not satisfy the random oracle assumption made in our formal security analysis. A production-level implementation would require replacing this with a certified CSPRNG. The threshold  $k$  was configured as a fixed parameter for each test run.
- iii) **Public Key Protocol Initialization:** An IND-CCA2 secure key pair  $(pk, sk)$  for the hybrid encryption’s public-key component  $\Pi$  is generated. We used the `cryptography` library to create an RSA key pair with a 2048-bit modulus, employing RSA-OAEP for encryption and decryption within  $\Pi$ . The protocol that we used to demonstrate the scheme is described in Section 7.3.

*Share Derivation:* The sender and receiver execute the same functions to derive the shares from the public cover text and the secret seed  $x_0$ . Our implementation, in the `generate_secure_x_values` and `text_to_shares` functions, proceeds as follows:

- i) **Abscissae Generation:** A sequence of abscissae  $(x_j)$  is generated by creating a hash chain:  $x_1 = H(x_0)$  and  $x_j = H(\text{encode}(x_{j-1}))$  for subsequent values, where `encode` is a byte-string conversion of the field element.
- ii) **Share Value Computation:** The PRNG, seeded with  $x_0$ , selects  $k - 1$  word indices from the cover text. The corresponding words are then hashed using SHA-256 (modulo  $p$ ) to produce the share values  $(s_j)$ . This results in a set of  $k - 1$  points  $\{(x_j, s_j)\}$ .

*Encryption:* Using  $k - 1$  shares from the derived set, a degree- $(k - 1)$  polynomial  $P(x)$  was interpolated with  $P(0) = m$  via Lagrange interpolation (implemented with the `galois` library). A fresh share  $(x_{\text{new}}, s_{\text{new}})$  was computed, and  $M = x_0 \| s_{\text{new}}$  was encrypted with RSA-OAEP to produce  $C_{\text{pub}}$ . A decoy message was encrypted with AES-CBC using  $s_{\text{new}}$  as the key, yielding  $C_{\text{sym}}$ .

*Decryption:* The receiver decrypts  $C_{\text{pub}}$  with the secret key  $sk$  to recover  $x_0$  and  $s_{\text{new}}$ , regenerates shares, and interpolates  $P(x)$  with the same  $k - 1$  shares plus  $(x_{\text{new}}, s_{\text{new}})$  to retrieve  $m$ .

Optimisations in the implementation included leveraging the `galois` library’s efficient finite field operations for polynomial interpolation and shared computations, as well as utilising the `cryptography` library’s optimised implementations of RSA-OAEP and AES-CBC, thereby minimising computational overhead.

### 7.3 Hiding the final share in a hybrid key exchange

The final share is embedded as part of a cryptographic protocol, in our case, the exchange of  $x_0$  as a “session key” using hybrid encryption: let the message to be encrypted under RSA be  $M = s_{\text{new}}$ , and let it be  $n = 256$  bit long. She computes  $C_{\text{pub}} = \text{RSA\_OAEP}_{pk}(M)$ , under Bob’s public key  $pk$ , and then treats the value of  $s_{\text{new}}$  as an AES key  $K$ . Selecting a random 128-bit  $IV$ , she encrypts a benign decoy payload  $D$  under AES-CBC with PKCS#7 padding, yielding  $C_{\text{sym}} = \text{AES\_CBC}_K(D, IV)$ . This step creates a decoy transcript of RSA-OAEP to conceal  $s_{\text{new}}$  as a session key for conventionally encrypted communication rather than part of a covert protocol. Alice transmits the pair  $(C_{\text{pub}}, C_{\text{sym}})$  to Bob. To the adversary,  $C_{\text{pub}}$  looks like a routine key-exchange message and  $C_{\text{sym}}$  like an ordinary encrypted document. Upon receiving these, Bob decrypts  $C_{\text{pub}}$  using his private key to retrieve  $s_{\text{new}}$ . Using the secrets  $k, x_0$  shared with Alice (beforehand), Bob generates the same shares as Alice and selects the same  $k - 1$  shares (this selection can be deterministically derived from  $x_0$ ). Finally, Bob reconstructs the secret  $m$  using the  $k - 1$  shares and the received  $s_{\text{new}}$ .

If questioned by an adversary, both Alice and Bob can claim they were exchanging the encrypted message  $C_{\text{sym}}$ , using standard public key cryptography to securely transmit the session key  $s_{\text{new}}$ .

### 7.4 Covertneess Analysis

The covertneess of our scheme stems from the legitimate appearance of the reference text  $T$ , the use of standard cryptographic primitives for the encrypted messages, and the decoy message  $D$  that appears to be the main encrypted payload.

The dual purpose of  $s_{\text{new}}$  is particularly important for covertneess. To an observer, the protocol appears to be a standard hybrid encryption method: a public key algorithm is used to securely transmit a symmetric key, which then encrypts the main message. This pattern matches legitimate encrypted communications, making Invisible Encryption indistinguishable from conventional secure messaging protocols.

An adversary cannot distinguish between the legitimate use of encryption for regular secure communication and our stealth protocol without breaking the underlying cryptographic primitives or obtaining access to the private keys of the participants.

### 7.5 Performance Analysis and Applications

We implemented the Invisible Encryption scheme in Python using the `galois` field library for finite field operations and the `cryptography` package for encryption primitives. Below, we present comprehensive performance metrics derived from our implementation, evaluated on a standard workstation (2.40 GHz Intel Core i5, 16 GB RAM) running Python 3.8.

Table 2: Detailed Performance Metrics for Core Operations

Operation	Mean Execution Time (ms)	Standard Deviation (ms)	Memory Usage (KB)
Field Initialization	1.07595	0.35538	225000.0
Secure x-value Generation ( $n = 5$ )	0.302410	0.50586	225000.0
Text-to-Shares Mapping ( $n = 5$ )	0.11942	0.37765	225000.0
New Share Creation ( $k = 3$ )	1.31597	0.43569	225000.0
Secret Reconstruction ( $k = 3$ )	1.24387	0.50298	225000.0
RSA Encryption (2048-bit)	247.46129	135.38926	225000.0
AES-CBC Encryption (decoy, 1KB)	304.03792	121.10442	225000.0

Table 3: Performance Metrics Across Different Threshold Configurations

Parameter	Share Generation (ms)	New Share Creation (ms)	Secret Reconstruction (ms)	Total Protocol Time (ms)	Comm. Overhead (bytes)
$k = 3, n = 5$	0.0	2.36959	1.58011	103.94971	544
$k = 5, n = 10$	1.65686	4.31780	3.72588	109.70056	544
$k = 7, n = 15$	1.57783	6.30636	9.57479	117.45898	544

Table 4: Decoy Message Encryption Performance (using  $s_{\text{new}}$  as session key)

Decoy Message Size	Encryption Time (ms)	Decryption Time (ms)	Ciphertext Size (bytes)
1 KB	194.77112	0.30829	1040
10 KB	249.74727	0.73943	10256
100 KB	250.28808	0.45835	102416
1 MB	257.22391	0.49147	1048592

The communication overhead remains constant at 544 bytes regardless of the threshold  $k$  and total shares  $n$ , as only  $x_0$  (32 bytes) and  $s_{\text{new}}$  (32 bytes) are transmitted, along with the RSA-encrypted payload (480 bytes for RSA-2048) and the encrypted decoy message which varies based on content size.

The dual use of  $s_{\text{new}}$  as both a share for secret reconstruction and an encryption key for the decoy message incurs no additional computational overhead, as the same value serves both purposes without requiring further processing. Our implementation of AES-CBC encryption using  $s_{\text{new}}$  as the key demonstrates performance comparable to standard implementations, with linear scaling for larger decoy messages. Our measurements show that even with a more complex setup ( $k = 7, n = 15$ ), the entire protocol executes in less than 400 ms, making it suitable for real-time applications. The communication overhead remains minimal and constant regardless of parameter choices, which is particularly valuable in bandwidth-constrained environments.

Invisible Encryption is particularly suitable for censorship-resistant communication in environments where encryption is monitored or prohibited, whistleblower protection allowing sensitive information to be transmitted covertly, diplomatic communications when revealing the existence of communication could have political implications, covert operations requiring maximum deniability, and privacy-preserving systems where regular encryption may draw unwanted attention.

The plausible deniability feature makes the system especially valuable in jurisdictions with key disclosure laws, where users may be legally compelled to reveal encryption keys. By providing a legitimate-looking decoy message encrypted with  $s_{\text{new}}$ , users can comply with such demands without revealing the existence of the covert channel.

## 7.6 Illustration with Ciphertext taken from this paper

We use this **section** to illustrate the scheme by taking **exactly** the first paragraph of this section as the ciphertext to embed a secret message inside. The additional share is embedded as a login token in the URL for a weblink that an adversary could **try** to open and be prompted for a password, but by the event of clicking on it, already having disclosed its attack attempt **to** Alice and Bob thus building in some very **simple** form of intrusion detection for an attacker that is unaware of the use of invisible encryption).





The above text consists of exactly  $L = 93$  words, with a maximum length of  $\leq n = 12$  characters. We let the implemented code run with a 256-bit prime  $p$ , using the seed  $x_0 = 6$  and  $k = 6$  shares, with  $PRNG(x) = H(x) \text{ MOD } 93$  as the pseudorandom number generator, where  $H$  is SHA256. The selected words, pointed to by the index sequence  $i_1, i_2, \dots, i_{k-1}$ , are **shown bold-printed** in the above paragraph (for illustration only), and the final share  $s_{new}$  is Base64-encoded into the “auth” token of the weblink inside the QR code.

### 7.7 Practical Considerations and Limitations

The implementation of Invisible Encryption, while functional, comes with practical considerations and limitations that must be acknowledged.

- **Implementation Complexity:** The protocol’s design requires the careful coordination of several cryptographic primitives, including polynomial interpolation, hybrid encryption, and pseudorandom sequence generation. This complexity can make secure implementation challenging and may introduce errors or vulnerabilities if not handled with expertise.
- **Security of the Secret Seed:** The security of the entire protocol hinges on the confidentiality of the secret seed  $x_0$ . Any leakage of this seed would completely compromise the system, allowing an adversary to reconstruct the secret message by identifying the correct shares. Therefore, protecting  $x_0$  through robust measures, such as physical or logical isolation on the user’s device, is of paramount importance.
- **Robustness against rephrasing:** The protocol is vulnerable against substitution of synonyms or rephrasing the cover text (unknowingly, or intentionally if the adversary aims to make the hidden message non-recoverable). In real-world scenarios, ambiguous words, idioms, or context-dependent meanings could be changed without affecting the carrier text itself, but produce a different interpolated polynomial and hence modify the secret up to non-recoverability.
- **Generalizability:** The prototype was tested in a controlled environment. Its performance and robustness have not been evaluated using large-scale, real-world datasets such as social media texts or system log files. This limits the generalizability of our findings to diverse, uncontrolled scenarios.

## 8 Conclusion and Future Work

We have introduced Invisible Encryption. This novel cryptographic protocol seamlessly integrates Shamir’s threshold secret sharing, steganographic embedding, and hybrid public-key encryption to facilitate covert communication in environments where traditional encryption is monitored or prohibited. By camouflaging a single share  $s_{new}$  within standard hybrid-encrypted traffic, our scheme ensures that the very act of secret transmission remains indistinguishable from ordinary ciphertext, while retaining complete IND-CCA2 security and efficient performance.

At its core, Invisible Encryption leverages a  $(k, n)$ -threshold sharing of the hidden payload: only one share is transmitted alongside the cover message, and the remaining  $k - 1$  shares are deterministically derived from a public cover text. The transmitted share  $s_{\text{new}}$  doubles as a symmetric key for an overt decoy message, yielding strong plausible deniability: even if compelled to reveal keys or plaintexts, participants can credibly claim the data exchanged was merely routine encrypted content.

Our rigorous security analysis demonstrates confidentiality, covertness, and detection-resistance under standard assumptions (e.g., RSA-OAEP and the secrecy of Shamir shares). A Python prototype, built on the `galois` and `cryptography` libraries, confirms that encryption and decryption complete in under 400 ms with negligible overhead, making the scheme practical for real-world scenarios such as whistleblowing, diplomatic messaging, or communications under repressive regimes.

Extending Invisible Encryption to multiple-message settings remains an important direction for future work. Naïve reuse of the same threshold key leaks the XOR of successive plaintexts; to mitigate this, one may:

- **Renew per-message keys:** derive a fresh  $(x_0, k)$  by hashing the previous ciphertext into the next seed.
- **Update cover text or embedding indexes:** rotate or refresh the public cover text, or embed the secret in higher-order polynomial coefficients to decorrelate shares.
- **Pre-encrypt whitening:** compress or otherwise uniformise the plaintext to resist Friedman-style statistical attacks.

Formal proofs of security under these enhancements and an evaluation of their performance and usability constitute compelling avenues for further research. The techniques presented here will inspire new approaches to secure, undetectable information exchange in an age of pervasive surveillance.



## References

1. TrueCrypt (Apr 2025), <https://en.wikipedia.org/w/index.php?title=TrueCrypt&oldid=1283747827>, page Version ID: 1283747827
2. Abdalla, M., Bellare, M., Rogaway, P.: The Oracle Diffie-Hellman Assumptions and an Analysis of DHIES. In: Naccache, D. (ed.) *Topics in Cryptology — CT-RSA 2001*. pp. 143–158. Springer, Berlin, Heidelberg (2001). [https://doi.org/10.1007/3-540-45353-9\\_12](https://doi.org/10.1007/3-540-45353-9_12)
3. Ahmad, S., Rass, S., Schartner, P.: False-Bottom Encryption: Deniable Encryption from Secret Sharing. *IEEE Access* pp. 1–1 (2023). <https://doi.org/10.1109/ACCESS.2023.3288285>, conference Name: IEEE Access
4. Barak, B., Goldreich, O., Impagliazzo, R., Rudich, S., Sahai, A., Vadhan, S., Yang, K.: On the (im)possibility of obfuscating programs. *J. ACM* **59**(2), 6:1–6:48 (May 2012). <https://doi.org/10.1145/2160158.2160159>, <https://dl.acm.org/doi/10.1145/2160158.2160159>
5. Benaloh, J.C.: Secret Sharing Homomorphisms: Keeping Shares of a Secret Secret (Extended Abstract). In: Odlyzko, A.M. (ed.) *Advances in Cryptology — CRYPTO’ 86*. pp. 251–260. Springer, Berlin, Heidelberg (1987). [https://doi.org/10.1007/3-540-47721-7\\_19](https://doi.org/10.1007/3-540-47721-7_19)
6. Bendlin, R., Damgård, I., Orlandi, C., Zakarias, S.: Semi-homomorphic Encryption and Multiparty Computation. In: Paterson, K.G. (ed.) *Advances in Cryptology – EUROCRYPT 2011*. pp. 169–188. Springer, Berlin, Heidelberg (2011). [https://doi.org/10.1007/978-3-642-20465-4\\_11](https://doi.org/10.1007/978-3-642-20465-4_11)
7. BLAKLEY, G.R.: Safeguarding cryptographic keys. In: 1979 International Workshop on Managing Requirements Knowledge (MARK). pp. 313–318 (Jun 1979). <https://doi.org/10.1109/MARK.1979.8817296>, <https://ieeexplore.ieee.org/document/8817296>, ISSN: 2164-0149

8. Bogdanov, D., Laur, S., Willemson, J.: Sharemind: A Framework for Fast Privacy-Preserving Computations. In: Jajodia, S., Lopez, J. (eds.) *Computer Security - ESORICS 2008*. pp. 192–206. Springer, Berlin, Heidelberg (2008). [https://doi.org/10.1007/978-3-540-88313-5\\_13](https://doi.org/10.1007/978-3-540-88313-5_13)
9. Boneh, D., Sahai, A., Waters, B.: Functional Encryption: Definitions and Challenges (2010), <https://eprint.iacr.org/2010/543>, publication info: Published elsewhere. Unknown where it was published
10. Canetti, R., Dwork, C., Naor, M., Ostrovsky, R.: Deniable Encryption. In: Kaliski, B.S. (ed.) *Advances in Cryptology — CRYPTO '97*. pp. 90–104. Springer, Berlin, Heidelberg (1997). <https://doi.org/10.1007/BFb0052229>
11. Chang, C.Y., Clark, S.: Practical Linguistic Steganography using Contextual Synonym Substitution and a Novel Vertex Coding Method. *Computational Linguistics* **40**(2), 403–448 (Jun 2014). [https://doi.org/10.1162/COLI\\_a\\_00176](https://doi.org/10.1162/COLI_a_00176), [https://doi.org/10.1162/COLI\\_a\\_00176](https://doi.org/10.1162/COLI_a_00176)
12. Chapman, M., Davida, G.: Hiding the hidden: A software system for concealing ciphertext as innocuous text. In: Han, Y., Okamoto, T., Qing, S. (eds.) *Information and Communications Security*. pp. 335–345. Springer, Berlin, Heidelberg (1997). <https://doi.org/10.1007/BFb0028489>
13. Cramer, R., Damgård, I.B., Nielsen, J.B.: *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, Cambridge (2015). <https://doi.org/10.1017/CB09781107337756>, <https://www.cambridge.org/core/books/secure-multiparty-computation-and-secret-sharing/4C2480B202905CE5370B2609F0C2A67A>
14. Cramer, R., Shoup, V.: Design and Analysis of Practical Public-Key Encryption Schemes Secure against Adaptive Chosen Ciphertext Attack (2001), <https://eprint.iacr.org/2001/108>, publication info: Published elsewhere. Unknown where it was published
15. Damgård, I., Pastro, V., Smart, N., Zakarias, S.: Multiparty Computation from Somewhat Homomorphic Encryption. In: Safavi-Naini, R., Canetti, R. (eds.) *Advances in Cryptology – CRYPTO 2012*. pp. 643–662. Springer, Berlin, Heidelberg (2012). [https://doi.org/10.1007/978-3-642-32009-5\\_38](https://doi.org/10.1007/978-3-642-32009-5_38)
16. Desmedt, Y., Frankel, Y.: Threshold cryptosystems. In: Brassard, G. (ed.) *Advances in Cryptology — CRYPTO' 89 Proceedings*. pp. 307–315. Springer, New York, NY (1990). [https://doi.org/10.1007/0-387-34805-0\\_28](https://doi.org/10.1007/0-387-34805-0_28)
17. Ding, J., Schmidt, D.: Rainbow, a New Multivariable Polynomial Signature Scheme. In: Ioannidis, J., Keromytis, A., Yung, M. (eds.) *Applied Cryptography and Network Security*. pp. 164–175. Springer, Berlin, Heidelberg (2005). [https://doi.org/10.1007/11496137\\_12](https://doi.org/10.1007/11496137_12)
18. Dodis, Y., Kiltz, E., Pietrzak, K., Wichs, D.: Message authentication, revisited. In: *Proceedings of the 31st Annual international conference on Theory and Applications of Cryptographic Techniques*. pp. 355–374. EUROCRYPT'12, Springer-Verlag, Berlin, Heidelberg (Apr 2012). [https://doi.org/10.1007/978-3-642-29011-4\\_22](https://doi.org/10.1007/978-3-642-29011-4_22), [https://doi.org/10.1007/978-3-642-29011-4\\_22](https://doi.org/10.1007/978-3-642-29011-4_22)
19. Feldman, P.: A practical scheme for non-interactive verifiable secret sharing. In: *Proceedings of the 28th Annual Symposium on Foundations of Computer Science*. pp. 427–438. SFCS '87, IEEE Computer Society, USA (Oct 1987). <https://doi.org/10.1109/SFCS.1987.4>, <https://doi.org/10.1109/SFCS.1987.4>
20. Garg, S., Gentry, C., Halevi, S., Raykova, M., Sahai, A., Waters, B.: Candidate Indistinguishability Obfuscation and Functional Encryption for all Circuits. In: *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*. pp. 40–49 (Oct 2013). <https://doi.org/10.1109/F0CS.2013.13>, <https://ieeexplore.ieee.org/document/6686139>, ISSN: 0272-5428
21. Goldreich, O., Micali, S., Wigderson, A.: How to play ANY mental game. In: *Proceedings of the nineteenth annual ACM symposium on Theory of computing*. pp. 218–229. STOC '87, Association for Computing Machinery, New York, NY, USA (Jan 1987). <https://doi.org/10.1145/28395.28420>, <https://dl.acm.org/doi/10.1145/28395.28420>
22. Herzberg, A., Jarecki, S., Krawczyk, H., Yung, M.: Proactive Secret Sharing Or: How to Cope With Perpetual Leakage. In: Coppersmith, D. (ed.) *Advances in Cryptology — CRYPTO' 95*. pp. 339–352. Springer, Berlin, Heidelberg (1995). [https://doi.org/10.1007/3-540-44750-4\\_27](https://doi.org/10.1007/3-540-44750-4_27)
23. Hoffstein, J., Pipher, J., Silverman, J.H.: NTRU: A ring-based public key cryptosystem. In: Buhler, J.P. (ed.) *Algorithmic Number Theory*. pp. 267–288. Springer, Berlin, Heidelberg (1998). <https://doi.org/10.1007/BFb0054868>

24. Komargodski, I., Naor, M., Yogev, E.: How to Share a Secret, Infinitely (2016), <https://eprint.iacr.org/2016/194>, publication info: A minor revision of an IACR publication in TCC 2016
25. Komlo, C., Goldberg, I.: FROST: Flexible Round-Optimized Schnorr Threshold Signatures. In: Selected Areas in Cryptography: 27th International Conference, Halifax, NS, Canada (Virtual Event), October 21-23, 2020, Revised Selected Papers. pp. 34–65. Springer-Verlag, Berlin, Heidelberg (Oct 2020). [https://doi.org/10.1007/978-3-030-81652-0\\_2](https://doi.org/10.1007/978-3-030-81652-0_2), [https://doi.org/10.1007/978-3-030-81652-0\\_2](https://doi.org/10.1007/978-3-030-81652-0_2)
26. Krawczyk, H.: Secret Sharing Made Short. In: Proceedings of the 13th Annual International Cryptology Conference on Advances in Cryptology. pp. 136–146. CRYPTO '93, Springer-Verlag, Berlin, Heidelberg (Aug 1993)
27. Lyubashevsky, V., Peikert, C., Regev, O.: On Ideal Lattices and Learning with Errors over Rings. J. ACM **60**(6), 43:1–43:35 (Nov 2013). <https://doi.org/10.1145/2535925>, <https://dl.acm.org/doi/10.1145/2535925>
28. McEliece, R.J.: A Public-Key Cryptosystem Based On Algebraic Coding Theory. Deep Space Network Progress Report **44**, 114–116 (Jan 1978), <https://ui.adsabs.harvard.edu/abs/1978DSNPR..44..114M>, aDS Bibcode: 1978DSNPR..44.114M
29. Pedersen, T.P.: Non-Interactive and Information-Theoretic Secure Verifiable Secret Sharing. In: Proceedings of the 11th Annual International Cryptology Conference on Advances in Cryptology. pp. 129–140. CRYPTO '91, Springer-Verlag, Berlin, Heidelberg (Aug 1991)
30. Rivest, R.L., Shamir, A., Adleman, L.: A method for obtaining digital signatures and public-key cryptosystems. Commun. ACM **21**(2), 120–126 (Feb 1978). <https://doi.org/10.1145/359340.359342>, <https://dl.acm.org/doi/10.1145/359340.359342>
31. Safaka, I., Fragouli, C., Argyraki, K.: Matryoshka: Hiding Secret Communication in Plain Sight (2016), <https://www.usenix.org/conference/foci16/workshop-program/presentation/safaka>
32. Shamir, A.: How to share a secret. Commun. ACM **22**(11), 612–613 (Nov 1979). <https://doi.org/10.1145/359168.359176>, <https://dl.acm.org/doi/10.1145/359168.359176>
33. Shor, P.W.: Algorithms for quantum computation: discrete logarithms and factoring. In: Proceedings of the 35th Annual Symposium on Foundations of Computer Science. pp. 124–134. SFCS '94, IEEE Computer Society, USA (Nov 1994). <https://doi.org/10.1109/SFCS.1994.365700>, <https://doi.org/10.1109/SFCS.1994.365700>
34. Shoup, V.: Practical threshold signatures. In: Proceedings of the 19th international conference on Theory and application of cryptographic techniques. pp. 207–220. EUROCRYPT'00, Springer-Verlag, Berlin, Heidelberg (May 2000)
35. Tyagi, N., Grubbs, P., Len, J., Miers, I., Ristenpart, T.: Asymmetric Message Franking: Content Moderation for Metadata-Private End-to-End Encryption (2019), <https://eprint.iacr.org/2019/565>, publication info: A major revision of an IACR publication in CRYPTO 2019
36. Watanabe, Y., Shikata, J.: Identity-Based Hierarchical Key-Insulated Encryption Without Random Oracles. In: Proceedings, Part I, of the 19th IACR International Conference on Public-Key Cryptography — PKC 2016 - Volume 9614. pp. 255–279. Springer-Verlag, Berlin, Heidelberg (Mar 2016). [https://doi.org/10.1007/978-3-662-49384-7\\_10](https://doi.org/10.1007/978-3-662-49384-7_10), [https://doi.org/10.1007/978-3-662-49384-7\\_10](https://doi.org/10.1007/978-3-662-49384-7_10)
37. Wayner, P.: Disappearing Cryptography: Information Hiding: Steganography and Watermarking. Morgan Kaufmann (Jun 2009)
38. Yao, A.C.: Protocols for secure computations. In: 23rd Annual Symposium on Foundations of Computer Science (sfcs 1982). pp. 160–164 (Nov 1982). <https://doi.org/10.1109/SFCS.1982.38>, <https://ieeexplore.ieee.org/document/4568388>, ISSN: 0272-5428

# Eliminating Exponential Key Growth in PRG-Based Distributed Point Functions

Marc Damie<sup>1,2</sup><sup>\*</sup>, Florian Hahn<sup>1</sup>, Andreas Peter<sup>3</sup>, and Jan Ramon<sup>2</sup>

<sup>1</sup> University of Twente, The Netherlands

<sup>2</sup> Inria, France

<sup>3</sup> Carl von Ossietzky Universität Oldenburg, Germany

**Abstract.** Distributed Point Functions (DPFs) enable sharing secret point functions across multiple parties, supporting privacy-preserving technologies such as Private Information Retrieval, and anonymous communications. While 2-party PRG-based schemes with logarithmic key sizes have been known for a decade, extending these solutions to multi-party settings has proven challenging. In particular, PRG-based multi-party DPFs have historically struggled with practicality due to key sizes growing exponentially with the number of parties and the field size. Our work addresses this efficiency bottleneck by optimizing the PRG-based multi-party DPF scheme of Boyle et al. (EUROCRYPT’15). By leveraging the honest-majority assumption, we eliminate the exponential factor present in this scheme. Our construction is the first PRG-based multi-party DPF scheme with practical key sizes, and provides key up to  $3\times$  smaller than the best known multi-party DPF. This work demonstrates that with careful optimization, PRG-based multi-party DPFs can achieve practical performances, and even obtain top performances.

**Keywords:** Distributed Point Function · Function Secret Sharing · Private Information Retrieval · Multi-Party Computations.

## 1 Introduction

Function Secret Sharing [2] is a cryptographic primitive enabling to share secret functions. In these protocols, a key dealer knowing a secret function  $f$  distributes  $p$  keys to different shareholder. Each shareholder can use its key to obtain a share of  $f(x)$ , without any communication between the shareholders.

Among all function families, schemes supporting point functions (i.e.,  $f(x) = \beta$  if  $x = \alpha$ , 0 otherwise) attracted a lot of attention thanks to their numerous applications notably in Private Information Retrieval (PIR) [7], in anonymous communications [6], in digital currencies [9], and machine learning [4]. These schemes are called “Distributed Point Functions” (DPF) [2,7].

To support these applications, there is a significant research incentive aiming to improve existing schemes, notably their key size. DPF efficiency is commonly evaluated based on the influence of the function domain size ( $N$ ) on the key size.

<sup>\*</sup> Corresponding author: [m.f.d.damie@utwente.nl](mailto:m.f.d.damie@utwente.nl)

For two- and three-party DPF, schemes based on PseudoRandom Generators (PRGs) provide logarithmic key sizes [2,9]. However, there is still a lot of active research to obtain similar key sizes for any arbitrary number of parties.

In multi-party DPF, three main approaches have emerged. First, elliptic-curve-based schemes [6,8] offer practical  $O(\sqrt{N})$  key sizes, but they require a non-linear share decoding. This non-linearity makes them incompatible with several key applications such as PIR. Second, Boyle et al. [2] presented a dishonest-majority scheme with  $O(\sqrt{N})$  key size, and Bunn et al. [5] an honest-majority scheme with  $O(\sqrt[4]{N})$  key size. Unfortunately, this asymptotic cost hides an exponential factors  $q^p$  (for output shares in  $\mathbb{F}_q$  and  $p$  parties). This factor makes these schemes impractical for any modulus  $q > 210$  (as detailed in Section 4). This problem was identified in existing works [1,8], but has not been solved *yet*.

Finally, Bunn et al. [5] proposed a third approach based on honest-majority to build an information-theoretic (IT) scheme with  $O(\sqrt{N})$  key size *and no exponential factor*. This scheme is the only multi-party scheme with practical key sizes supporting all applications. Even though this scheme is practical, solving the efficiency issues of the other schemes could lead to even better performances. As PRGs have lead to logarithmic key sizes in two and three-party schemes, optimizing multi-party PRG-based schemes could be promising.

*Our Contributions* Our paper optimizes the PRG-based DPF proposed by [2]. Our optimized scheme avoids the exponential factors present in [2] using the honest-majority assumption; obtaining a key size of  $O(\sqrt{N} \cdot \binom{p}{m+1} \log q)$  instead of  $O(\sqrt{N} q^{\frac{p-1}{2}} \log q)$ . Our benchmark shows that our scheme is the first multi-party PRG-based scheme with practical key sizes. We even provide keys up to  $3\times$  smaller than the best performing DPF (i.e., the IT DPF by [5]).

*Notations* Let  $p$  be the number of parties/shareholders,  $m$  be the number of corrupted parties. Like most FSS works [2,3,5,7,8], we focus on semi-honest adversaries: follow the protocol and infer *passively* secret information.

Let  $\mathbb{F}_q$  be a prime field,  $N$  be the function domain size,  $1^\lambda$  a security parameter, and  $\llbracket x \rrbracket_i$  be the  $i$ -th share of the secret  $x$ . Let  $\nu = \lceil \sqrt{N} \rceil$  and  $C = \binom{p}{m+1}$ . Let  $G : \{0, 1\}^\lambda \rightarrow \mathbb{G}^\nu$  be a PRG, and  $\mathbb{G}$  an Abelian group.

## 2 Background

Function secret sharing (FSS) [2] enables sharing secret functions between  $p$  parties. Each FSS scheme can share function from a specific function family.

A function family  $\mathcal{F}$  [1] is a pair  $(P_{\mathcal{F}}, E_{\mathcal{F}})$  where  $P_{\mathcal{F}} \subseteq \{0, 1\}^*$  is a collection of function descriptions  $\hat{f}$ , and  $E_{\mathcal{F}}: P_{\mathcal{F}} \times \{0, 1\}^* \rightarrow \{0, 1\}^*$  is a polynomial-time algorithm defining the function described by  $\hat{f}$ ; i.e.,  $f(x) = E_{\mathcal{F}}(\hat{f}, x)$ . All functions within a family share the same domain  $\mathcal{X}$  and output space  $\mathcal{Y}$ .

Due to their applications notably in PIR [7] and anonymous communications [6], the most studied function family has been point functions [2,3,5,6,7];

functions  $f$  such that  $f(x) = \beta$  if  $x = \alpha$ , and  $f(x) = 0$  otherwise. For point functions, the function description is the tuple  $(\alpha, \beta) \in P_{\mathcal{F}}$ . Schemes supporting point functions are called “Distributed Point Functions” (DPF).

For a function family  $\mathcal{F}$ , we define a  $p$ -party FSS scheme using 3 algorithms:

- Gen :  $\mathbb{N} \times P_{\mathcal{F}} \rightarrow \mathcal{K}^p$  takes as input a security parameter  $1^\lambda \in \mathbb{N}$  and a function description  $\hat{f} \in P_{\mathcal{F}}$ , and outputs  $p$  keys  $k_1, \dots, k_p$ .
- Eval :  $\mathcal{K} \times \mathcal{X} \rightarrow \mathbb{G}$  takes as input  $k_i$  and a point  $x \in \mathcal{X}$ , outputs a share of  $f(x)$  that we denote as  $\llbracket f(x) \rrbracket_i$ .
- Decode :  $\mathbb{G}^p \rightarrow \mathcal{Y}$  takes as input the shares  $\{\llbracket f(x) \rrbracket_1, \dots, \llbracket f(x) \rrbracket_p\}$  and outputs the secret  $f(x)$ .

**Definition 1 (Correctness [2]).** A scheme (Gen, Eval, Decode) is correct if, for any function  $f \in \mathcal{F}$  and point  $x \in \mathcal{X}$ , we have:

$$\Pr[\text{Decode}(\text{Eval}(k_1, x), \dots, \text{Eval}(k_p, x)) = f(x)] = 1$$

with  $k_1, \dots, k_p \leftarrow \text{Gen}(1^\lambda, \hat{f})$

**Definition 2 (Privacy [1]).** Let  $\text{Leak} : \{0, 1\}^* \rightarrow \{0, 1\}^*$  be a function specifying the allowable leakage. A scheme (Gen, Eval, Decode) is private if, for every set of  $m$  corrupted parties  $S \subseteq \{1 \dots p\}$ , there exists a PPT algorithm Sim (simulator), s.t. for any sequence of function descriptions  $(\hat{f}_1, \hat{f}_2, \dots)$  of size polynomial in  $\lambda$ , the outputs of Real and Ideal are computationally indistinguishable:

- Real( $1^\lambda$ ) :  $(k_1, \dots, k_p) \leftarrow \text{Gen}(1^\lambda, \hat{f}_\lambda)$ ; Output  $(k_i)_{i \in S}$
- Ideal( $1^\lambda$ ) : Output Sim( $1^\lambda, \text{Leak}(\hat{f}_\lambda)$ )

*DPF by [2]* To build their multi-party DPF scheme under dishonest majority ( $m < p$ ), Boyle et al. [2] represented the domain  $\{1, \dots, N\}$  as a  $\nu \times \nu$  grid (with  $\nu = \lceil \sqrt{N} \rceil$ ). This grid is full of zeros except on the cell  $(\gamma_*, \delta_*)$ , with  $\alpha = \gamma_*\nu + \delta_*$ .

For each row  $\gamma \in \{1, \dots, \nu\}$ , the Gen algorithm samples  $q^{p-1}$  random  $\lambda$ -bit random seeds  $s_{\gamma,1} \dots s_{\gamma,q^{p-1}}$ . For each seed  $s_{\gamma,j}$ , the algorithm generates additive shares of a coefficient  $a_{\gamma,j}$ :  $\llbracket a_{\gamma,j} \rrbracket_1, \dots, \llbracket a_{\gamma,j} \rrbracket_p$  (i.e., one share per DPF key). The coefficient is defined as follows  $a_{\gamma,j} = 1$  if  $\gamma = \gamma_*$ , 0 otherwise. Finally, it sets a “correction word”  $W \in (\mathbb{F}_q)^\nu$  such that  $W + \sum_{i=1}^{q^{p-1}} G(s_{\gamma_*,j}) = e_{\delta_*} \cdot \beta$  (with  $e_\delta$  a unit vector equal to 1 on index  $\delta$ , 0 otherwise). Each key  $k_i$  contains the correction word, their share of the coefficients  $\llbracket a_{\gamma,j} \rrbracket_i$  (for all rows  $\gamma$  and all  $j \in \{1, \dots, q^{p-1}\}$ ), and it contains all the seeds  $s_{\gamma,j}$  for which  $\llbracket a_{\gamma,j} \rrbracket_i \neq 0$ . This last condition (on the seed inclusion in a key) ensures that there is at least one seed unknown to an adversary composed of  $p-1$  out of  $p$  parties.

The Eval algorithm represents the input  $x$  as a tuple  $(\gamma, \delta)$ , expands the corresponding seeds  $s_{\gamma,j}$ , multiplies the expanded seeds with the corresponding shared coefficient  $\llbracket a_{\gamma,j} \rrbracket_i$ , sums everything with the correction word  $W$ , and the share  $\llbracket f(x) \rrbracket_i$  is on the  $\delta$ -th index of the sum vector:

$$\llbracket f(x) \rrbracket_i = v[\delta] \text{ with } v = W + \sum_j \llbracket a_{\gamma,j} \rrbracket_i \cdot G(s_{\gamma,j})$$

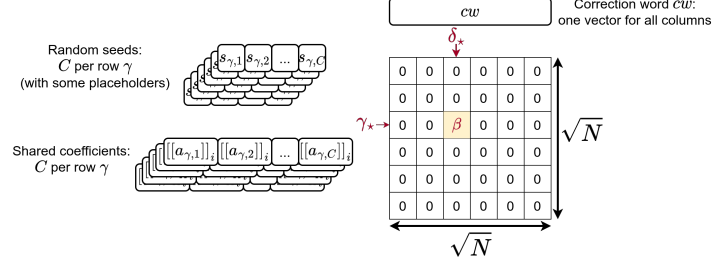


Fig. 1: Structure of our DPF keys

The Decode algorithm is a basic additive share decoding:  $\sum_i \llbracket f(x) \rrbracket_i = f(x)$ .

As we optimize [2], Algorithm 1 (describing our scheme) follows roughly the same structure as their scheme. The *only difference* is the matrix sampling in Lines 8 and 9. We then refer to Algorithm 1 for more details.

An element could surprise the reader: the number of random seeds  $\nu \times q^{p-1}$ . Such a large number is necessary, so an adversary cannot infer information about the secret function based on the distribution of the shares  $\llbracket a_{\gamma,j} \rrbracket_i$ . These shares can be structured as matrix shares  $A_\gamma$  with  $A_\gamma[i, j] = \llbracket a_{\gamma,j} \rrbracket_i$ . To preserve function privacy, each matrix must contain all combination of additive shares summing to 1 (if  $\gamma = \gamma_*$ ) or 0 (if  $\gamma \neq \gamma_*$ ). If the matrices do not contain all possible combinations, an adversary (owning up to  $p - 1$  keys out of  $p$ ) can recover  $\gamma_*$  based on the share distribution [2]. Since there exists  $q^{p-1}$  combinations of  $p$  shares of 0 (resp. of 1) in  $\mathbb{F}_q$ , the key generator must sample  $q^{p-1}$  random seeds.

### 3 Our honest-majority scheme

The main scalability bottleneck in [2] lies in the size of the matrices of shares, we improve their scheme by eliminating its dependence on the field size; thanks to the honest-majority assumption ( $m < p/2$ ). While Boyle et al. [2] had assumed a dishonest majority ( $m < p$ ), the honest-majority assumption enables us to redesign the matrices, resulting in more compact keys. Algorithm 1 presents our multi-party DPF scheme, and Figure 1 an overview of our DPF keys.

Our matrix sampling (described in the function `MatrixOfShares` of Algorithm 1) generates a matrix  $A$  that distributes shares among all possible combinations of  $m + 1$  parties out of  $p$ . For each combination  $S_j$  of  $m + 1$  parties, the function samples  $m + 1$  shares of the secret coefficient  $a$  (using additive secret sharing) and assigns each share to the cell of  $A$ ;  $A[i, j] = \llbracket a \rrbracket_i$  if  $i \in S_j$ , 0 otherwise. The secret coefficient  $a$  is either set to 0 or 1 depending on whether the value  $x$  being evaluated matches  $\alpha$  (the non-zero point).

While in [2], each matrix of shares has  $p$  rows and  $q^{p-1}$  columns, our construction produces matrices with  $p$  rows and  $\binom{p}{m+1}$  columns. As in [2], we sample one random seed per column. The  $i$ -th key contains the  $i$ -th row of the matrix and includes the  $j$ -th seed if the corresponding cell  $A[i, j]$  is not null.



---

**Algorithm 1** Honest-majority DPF scheme adapted from [2].

---

```

1: function MATRIXOFSHARES( $a$ )
2:   Initialize  $A$  an  $p \times C$  matrix with zeros and the counter  $k$  to 1.
3:   for each set of parties  $S_j$  in the set of all combinations of  $m + 1$  of  $p$  do
4:     Sample  $m + 1$  shares of the value  $a$ :  $\{\llbracket a \rrbracket_{1,i} \dots \llbracket a \rrbracket_{m+1,i}\}$ .
5:     for each  $i$  in  $S_j$  do  $A[i, j] \leftarrow \llbracket a \rrbracket_{k,i}$  and increment  $k$ .
   return  $A$ 

6: function Gen( $\alpha, \beta, p, m, 1^\lambda$ )
7:   Represent  $\alpha$  as a pair  $\alpha = (\gamma_*, \delta_*)$  with  $\gamma_*, \delta_* \in \{0 \dots \nu\}$ .
8:   Sample  $A_1, \dots, A_\nu$  s.t. for all  $\gamma \neq \gamma_*$ ,  $A_\gamma \leftarrow \text{MatrixOfShares}(0)$ .
9:   Sample  $A_{\gamma_*} \leftarrow \text{MatrixOfShares}(1)$ .
10:  Choose randomly and independently  $\nu \cdot C$  seeds  $s_{1,1}, \dots, s_{\nu,C} \in \{0, 1\}^\lambda$ .
11:  Set the correction words  $W \in \mathbb{G}^\nu$  s.t.  $W + \sum_{j=1}^C G(s_{\gamma_*,j}) = e_{\delta_*} \cdot \beta$ .
12:  for  $i \in \{1 \dots p\}, j \in \{1 \dots C\}, \gamma \in \{1 \dots \nu\}$  do
13:    if  $A_\gamma[i, j] \neq 0$  then  $\sigma_{i,\gamma,j} \leftarrow (s_{\gamma,j}, A_\gamma[i, j])$ .
14:    else  $\sigma_{i,\gamma,j} \leftarrow (0, 0)$ .  $\triangleright$  Receives no seed and no coefficient share
15:  Set  $\sigma_{i,\gamma} \leftarrow (\sigma_{i,\gamma,1} || \dots || \sigma_{i,\gamma,C})$  for all  $1 \leq i \leq p, 1 \leq \gamma \leq \nu$ .
16:  return  $(k_1, \dots, k_p)$  with  $k_i = (\sigma_{i,1} || \dots || \sigma_{i,\nu} || W)$  for all  $1 \leq i \leq p$ .

17: function Eval( $k_i, x$ )
18:  Represent  $x$  as a pair  $x = (\gamma, \delta)$  with  $\gamma, \delta \in \{0 \dots \nu\}$ .
19:  Parse  $k_i = ((s_{1,1}, A_1[i, 1]) || \dots || (s_{1,C}, A_1[i, C]) || \dots || (s_{\nu,C}, A_\nu[i, C]) || W)$ .
20:  return  $y_i[\delta]$  with  $y_i \leftarrow A_\gamma[i, 1] \cdot W + \sum_{j=1}^C A_\gamma[i, j] \cdot G(s_{\gamma,j})$ .

21: function Decode( $(\llbracket y \rrbracket_1, \dots, \llbracket y \rrbracket_p)$ ) return  $\sum_{i=1}^p \llbracket y \rrbracket_i$ .
```

---

As in [2], it is necessary that (for any given  $\gamma$ ) at least one seed  $s_{\gamma,j}$  remains unknown to the adversary. With all the seeds, they could unmask the correction word  $W$  and recover  $\beta$ . Our scheme provides each seed to  $m + 1$  parties, so *under honest majority*, at least one combination of  $m + 1$  out of  $p$  parties contains only honest agents. Thus, there is at least one seed unknown to an adversary.

Our optimization cannot be extended to dishonest majority. Indeed, with  $m > p/2$ , the adversary would know all the seeds because there would be at least one corrupted party in each combination of  $m + 1$  out of  $p$  parties.

Based on these intuitions, we can consider the following security theorem:

**Theorem 1.** *Let  $\lambda \in \mathbb{N}$ ,  $N, p \in \mathbb{N}$ , then the tuple (Gen, Eval, Decode) as described in Algorithm 1 is an FSS scheme for the family of all point functions with  $\alpha \in \{1, \dots, N\}$  and any  $\beta \in \mathbb{F}_q$ .*

*Assuming that there exists a secure PRG, then this scheme is correct and private against at most  $m$  semi-honest parties with  $m < p/2$ .*

*Proof.* As we modified only slightly the scheme of [2] (i.e., redesigned the matrix of shares based on the honest-majority assumption), our proof follows the same structure as theirs. We provide a proof sketch, and refer to [2] for more details.

The correctness can be verified by an easy arithmetic exercise considering successively three cases: (1)  $\gamma \neq \gamma_*$ , (2)  $\gamma = \gamma_*$  and  $\delta \neq \delta_*$ , and (3)  $\gamma = \gamma_*$  and  $\delta = \delta_*$ . Using the  $\sqrt{N} \times \sqrt{N}$  grid (illustrated in Figure 1), we represent any input  $x$  as  $(\gamma, \delta)$  and  $\alpha$  as  $(\gamma_*, \delta_*)$ .

For privacy, we must show that there exists a simulator that outputs samples from a distribution that is computationally indistinguishable from the distribution of the real DPF keys. We propose to study separately: the random seeds  $s_{\gamma,j}$ , the correction word  $W$ , and the coefficient shares  $A_\gamma[i, j]$ .

The simulation is straightforward: for each (simulated) seed, sample a random seed  $\widetilde{s}_{\gamma,j}$ ; for the correction word, sample a random vector  $\widetilde{W}$ ; for the coefficients, for each  $S_j$  combination of  $m+1$  parties of  $p$ , for each  $i$  in  $S_j$ , sample a random value and store it in  $\widetilde{A}_\gamma[i, j]$  (the rest of  $\widetilde{A}_\gamma$  is null). The simulator can return “simulated” keys based on these elements.

Since both the real and simulated seeds are randomly sampled, the simulators output distribution ( $\widetilde{s}_{\gamma,j}$ ) is computationally indistinguishable from that induced by the distribution of a single output of Gen.

The correction word  $W$  is a secret vector (i.e.,  $e_{\delta_*} \cdot \beta$ ) masked with the output of  $\binom{p}{m+1}$  seeded PRGs. Remember that a key  $k_i$  contains a seed  $s_{\gamma,j}$  only if the  $i$  is part of the  $j$ -th combination of  $m$  out of  $p$  parties. So, under honest majority, there is always at least one seed unknown to the adversary controlling  $m$  out of  $p$  parties. Hence, the correction word  $W$  is computationally indistinguishable from the randomly sampled  $\widetilde{W}$ , because it is masked with the output of (at least) one PRG seeded with a seed unknown to the adversary [6].

Finally, each coefficient is shared between  $m+1$  parties, so an adversary controlling  $m$  parties cannot distinguish real shares from random values  $\widetilde{A}_\gamma[i, j]$  provided by the simulator.  $\square$

*Key size optimization:* Instead of using a  $\sqrt{N} \times \sqrt{N}$  grid, we should use a grid with  $\sqrt{N}(C)^{-1}$  rows and  $\sqrt{N} \cdot C$  columns, with  $C = \binom{p}{m+1}$ . This “non-square” grid leverages the fact that each row requires  $C$  seeds, and a unique vector  $W$  for all columns. Thanks to this trick, we obtain a better key size:  $O(\sqrt{N} \cdot C \log q)$ .

*Extension to comparison functions* Boyle et al. [2] presented a simple adaptation of their DPF to support comparison functions; functions such that  $f(x) = \beta$  when  $x \leq \alpha$ , 0 otherwise. These schemes have applications notably in machine learning [4,8]. We can naively reuse our optimization on this other scheme.

## 4 Key size comparison

This section compares our optimized scheme to existing schemes in order to identify asymptotic and practical key size reductions.

We focus our comparison on schemes supporting all possible DPF applications; excluding schemes based on elliptic curves that support a limited number of applications due to their non-linear decoding [8] (e.g., do not support PIR).

*Asymptotic* Our scheme provides a key size of  $O(\sqrt{N \cdot \binom{p}{m+1}} \log q)$ , which is clearly better than the key size of [2] (i.e.,  $O(\sqrt{N} q^{\frac{p-1}{2}} \log q)$ ).

As Bunn et al. [5] built an honest-majority scheme with  $O(\sqrt[4]{N})$  key size upon [2], we can comment how we distinguish from them. Their paper does not modify [2], but combine it with replicated secret sharing to reduce the dependency on  $N$  (i.e.,  $O(\sqrt[4]{N})$  instead of  $O(\sqrt{N})$ ). However, their approach worsened the exponential factors already in [2]; as shown in our benchmark below. On the contrary, we leverage the honest-majority scheme to avoid exponential factors present in [2], but we maintained the same dependence on  $N$ .

The IT DPF by [5] has a key size comparable to ours:  $O(\sqrt{N} \cdot \binom{p}{m+1} \log q)$ . However, our approach saves a factor  $\sqrt{\binom{p}{m+1}}$  compared to them, yielding significant key size reductions in practice.

While these asymptotic comparisons are informative, they are often insufficient to assess practical performance. Bunn et al. [5] exemplify this problem: although they substantially reduced the dependence on the domain size  $N$ , they kept exponential factors without providing any concrete efficiency analysis. Therefore, we present a detailed comparison based on exact key sizes to offer a more accurate assessment.

*Exact* As we aimed to avoid the exponential factor  $q^p$  (for outputs in  $\mathbb{F}_q$ ), we start by studying the dependency on  $q$ . Figure 2 compares key sizes for varying prime moduli. Our benchmark includes a curve “Trivial scheme” corresponding to the most trivial DPF: sharing the function truth table (i.e.,  $O(N)$  key size). This curve serves as baseline to identify impractical solutions. For any  $q > 5$ , the PRG-based solutions [2,5] have keys orders of magnitude larger than those of this trivial solution, while the IT DPF of [5] and ours are below.

Recently, Boyle [1] showed that, for a composite modulus  $m = q_1 q_2 \dots q_l$ , we can use the Chinese Remainder Theorem (CRT) to replace  $q^p$  with  $\sum q_i^p$ . Figure 2a compares key sizes for arbitrary moduli, but the variations make the figure poorly readable. Instead, Figure 2c compares key sizes for primorial moduli; a primorial is the product of the first  $n$  primes. Primorials are the best case of this CRT trick as they provide composite moduli with the smallest primes possible.

Even with the CRT trick, the existing PRG-based schemes [2,5] provides key sizes larger than the trivial scheme for any modulus above 210. As their key sizes are impractical, we exclude these schemes from our other figures to focus on the comparison of our scheme to practical schemes.

Figure 2c also shows that the key size of the IT scheme of [5] grows faster with the modulus than ours. This phenomenon is explained by the fact that our key size is dominated by components conditioned by a security parameter that is independent of the modulus.

Figure 3a compares the practical schemes for varying function domain sizes. **Our key size is 2.4× smaller than the best existing scheme.** Moreover, Figure 3b shows that **our scheme has a better scaling with the number of parties** thanks to the factor  $\sqrt{\binom{p}{m+1}}$  identified in the asymptotic comparison.

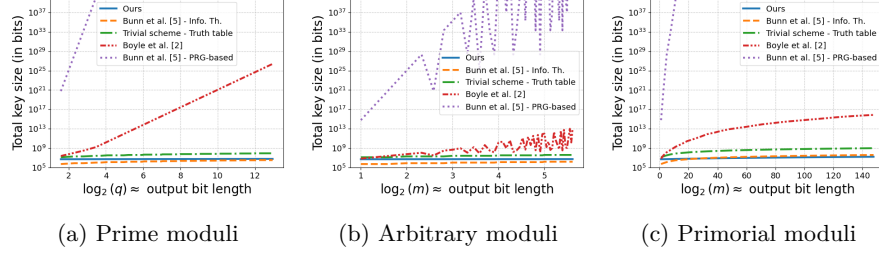


Fig. 2: Key size of various DPF schemes for varying moduli ( $p = 7$ ).

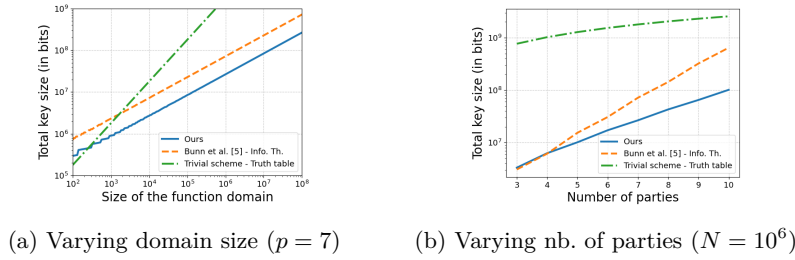


Fig. 3: Key size of the most efficient DPF schemes.

**Conclusion** Our optimization based on the honest-majority assumption transformed a PRG-based DPF [2] with impractical key sizes into the DPF with the smallest key sizes. Our work proves that, like in two- and three-party schemes, PRG is a promising primitive to build multi-party DPF with compact keys.

**Acknowledgments.** This work was supported by the Netherlands Organization for Scientific Research (De Nederlandse Organisatie voor Wetenschappelijk Onderzoek) under NWO:SHARE project [CS.011].

## References

1. Boyle, E.: Function Secret Sharing and Homomorphic Secret Sharing (2022)
2. Boyle, E., Gilboa, N., Ishai, Y.: Function Secret Sharing. In: Advances in Cryptology - EUROCRYPT 2015 (2015). [https://doi.org/10.1007/978-3-662-46803-6\\_12](https://doi.org/10.1007/978-3-662-46803-6_12)
3. Boyle, E., Gilboa, N., Ishai, Y.: Function Secret Sharing: Improvements and Extensions. In: Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security (Oct 2016). <https://doi.org/10.1145/2976749.2978429>
4. Boyle, E., Gilboa, N., Ishai, Y.: Secure Computation with Preprocessing via Function Secret Sharing. In: Theory of Cryptography (2019). [https://doi.org/10.1007/978-3-030-36030-6\\_14](https://doi.org/10.1007/978-3-030-36030-6_14)
5. Bunn, P., Kushilevitz, E., Ostrovsky, R.: CNF-FSS and Its Applications. In: Public-Key Cryptography 2022 (2022). [https://doi.org/10.1007/978-3-030-97121-2\\_11](https://doi.org/10.1007/978-3-030-97121-2_11)

6. Corrigan-Gibbs, H., Boneh, D., Mazières, D.: Riposte: An Anonymous Messaging System Handling Millions of Users. In: 2015 IEEE Symposium on Security and Privacy (May 2015). <https://doi.org/10.1109/SP.2015.27>
7. Gilboa, N., Ishai, Y.: Distributed Point Functions and Their Applications. In: EUROCRYPT 2014 (2014). [https://doi.org/10.1007/978-3-642-55220-5\\_35](https://doi.org/10.1007/978-3-642-55220-5_35)
8. Kumar, C., Patranabis, S., Mukhopadhyay, D.: Compact Key Function Secret Sharing with Non-linear Decoder. IACR Communications in Cryptology **1**(2) (Jul 2024). <https://doi.org/10.62056/a3c3c3w9p>
9. Zyskind, G., Yanai, A., Pentland, A.S.: High-Throughput Three-Party DPFs with Applications to ORAM and Digital Currencies. In: Proceedings of the 2024 ACM SIGSAC Conference on Computer and Communications Security (2024)

# A Pseudo-Inverse Matrix-Based LDP for High-Dimensional Data

Hiroaki Kikuchi<sup>1</sup>[0000–0002–0903–8430]

School of Interdisciplinary Mathematical Sciences, Meiji University.  
4-21-1 Nakano, Tokyo 164-8525, Japan. [kikn@meiji.ac.jp](mailto:kikn@meiji.ac.jp)

**Abstract.** We study local differential privacy (LDP) protocols that estimate the statistics of a group while preserving individual data privacy. One of the challenges for LDP is dealing with high-dimensional data, which is common in the medical domain and can incur a large privacy budget that grows with dimensionality. In 2018, Zhang et al. presented a novel LDP scheme, CALM (Consistent Adaptive Local Marginal), that could estimate the joint probability distribution of high-dimensional data via a set of lower-dimensional marginals, called “views.” The process involved entropy maximization with a convex optimization algorithm. However, the entropy maximization process may fail if the original data is over-randomized. We therefore propose a simple method that addresses this estimation issue using a pseudo-inverse matrix. We evaluate the accuracy of our estimation method in terms of the size of the views and frequency predictions.

**Keywords:** differential privacy, multi-dimensional data

## 1 Introduction

Local differential privacy (LDP) has been utilized in various privacy-enhancing applications. For instance, Erlingsson et al. proposed an LDP algorithm called the randomized aggregatable privacy-preserving ordinal response [9]. This algorithm is employed by Google Chrome to gather user data while ensuring privacy.

The dimensionality problem in LDP refers to the challenge of handling high-dimensional data efficiently and accurately while preserving privacy and has two main aspects. (a) *Increased Noise*. In LDP, each data item is perturbed independently to ensure privacy, which often involves adding noise to the data. As the dimensionality of the data increases, the amount of noise added to each dimension also increases, leading to a significant reduction in the accuracy of the aggregated data. (b) *Scalability*. Using high-dimensional data requires complex mechanisms to ensure that privacy is maintained for each dimension, and the aggregated domain can grow exponentially. This complexity can lead to LDP mechanisms becoming less scalable and more computationally intensive, making it difficult to handle large datasets efficiently.

A number of studies have addressed the dimensionality issue in LDP. Domingo-Ferrer et al. [1] introduced a clustering-based randomized response method. Ren

et al. [2] developed a technique called LoPub, which integrates Lasso regression with the Expectation-Maximization (EM) algorithm. Wang et al. [3] proposed using the Gaussian copula to enhance accuracy. Jiang et al. [4] introduced Wasserstein autoencoders as a solution.

The Consistent Adaptive Local Marginal (CALM) algorithm [6] was designed to address the challenges of high-dimensional data in the context of LDP. It aimed to improve the efficiency and accuracy of data collection and analysis while adhering to privacy constraints. It has three main advantages.

- *Communication Efficiency.* By focusing on a subset of the dimensions (the  $\ell$ -way marginals), rather than all dimensions, the amount of data that needs to be transmitted is reduced. The algorithm can then aggregate the local marginal statistics, which are simpler to compute and require less communication.
- *Marginal Aggregation.* By focusing on the marginal distributions of the data, CALM can provide accurate estimates without needing to handle the full high-dimensional joint distribution (the  $k$ -way marginals).
- *Adaptive Sampling.* CALM uses a sampling strategy to divide the whole population into  $\binom{k}{\ell}$  smaller subsets, where  $\ell$  is the size of subsets sampled over  $k$  attributes. This helps conserve the privacy budget and improve overall utility.

Despite its advantages, the CALM approach has a notable drawback: it relies on off-the-shelf convex optimization tools to solve the constraint problems. (It is worth noting that the central server aims to estimate  $k$ -way marginals based on a given set of  $\ell$ -way marginals. However, the available  $\ell$ -way marginals may be insufficient to uniquely determine the  $k$ -way marginals.) This dependency introduces its own challenges. In some cases, the convex optimization tool may fail to determine unique solution due to the nature of the noise added to ensure privacy enhancement. Given too strict or inconsistent constraints, it cannot find a solution.

To address this drawback of CALM, we investigate the required number of constraints for target dimensions. We introduce a new LDP algorithm that employs a *pseudo-inverse matrix* [12] to solve the constraints and estimate high-dimensional marginals, eliminating the need for convex optimization tools. A useful property of the pseudo-inverse is that it exists for any given matrix. Therefore, without having to use a potentially unstable convex optimization, we can estimate the  $k$ -way marginals using some lower-dimensional  $\ell$ -way marginals. Our scheme offers three main advantages:

- it is stable and consistently provides a solution to the given constraints,
- it estimates marginals as accurate as using the CALM,
- it is simple and easy to implement. Pseudo-inverse matrix is simply defined and many libraries are available.

Using open data, we conducted experiments to measure the accuracy of the proposed algorithm in Section 5. In Section 3.1, we clarify the necessary conditions for the number of views to have uniquely determined solutions. If the

domain is too large, it is a challenge to find high-dimensional marginals from low-dimensional views.

## 2 Preliminaries

### 2.1 Problem Definition

Let  $m$  be a number of attributes for multi-dimensional data. Let  $\Omega_i$  be the domain of the  $i$ -th attribute, and  $\Omega = \Omega_1 \times \cdots \times \Omega_m$  be the set of  $m$  domains. Let  $n$  be the number of users who each have  $m$  private input values,  $x_i^1, \dots, x_i^m$ . The users use a randomized algorithm to perturb these values and submit the perturbed values  $y_i^1, \dots, y_i^m$  to a data curator. Let  $(Y^1, \dots, Y^m)$  be the  $m$ -dimensional (columns) data of  $n$  records (rows).

Given a subset of  $(Y^1, \dots, Y^m)$  of size  $k$  such that  $k \leq m$ , we (as the data curator) try to estimate the  $k$ -dimensional joint probability  $\Phi^k$  as accurately as possible.

### 2.2 LDP

A good randomized algorithm  $\Psi$  is required to satisfy the following property.

**Definition 1 ( $\epsilon$ -local differential privacy).** An algorithm  $\Psi$  satisfies  $\epsilon$ -local differential privacy, where  $\epsilon \geq 0$ , if and only if for any inputs  $x_1, x_2 \in \Omega$ , we have

$$\forall T \in \text{Range}(\Psi) \quad \Pr[\Psi(v_1) \in T] \geq e^\epsilon \Pr[\Psi(v_2) \in T].$$

### 2.3 Generalized Randomized Response (GRR)

The GRR [8] is the generalized version of a randomized response. Let  $P$  be the

$(d \times d)$  randomizing matrix  $P = \begin{pmatrix} p_{11} & \cdots & p_{1d} \\ \vdots & \ddots & \vdots \\ p_{d1} & \cdots & p_{dd} \end{pmatrix}$ , where  $p_{uv}$  is the conditional probability that output variable  $Y$  takes  $v$ , given input variable  $X$  is  $u$ , i.e.,  $p_{uv} = \Pr(Y = v | X = u)$ . Note that  $p_{i1} + \cdots + p_{id} = 1$  for  $i = 1, \dots, d$ . According to  $P$ , perturbing input  $X$  to  $Y = (y_1, \dots, y_n)$  gives

$$y_i = \begin{cases} x_i & \text{with } p = p_{ii} = \frac{e^\epsilon}{e^\epsilon + w - 1}, \\ v \in \Omega - \{x_i\} & \text{with } q = p_{ij} = \frac{1}{e^\epsilon + w - 1}. \end{cases},$$

where  $w = |\Omega|$ , and GRR satisfies  $\epsilon$ -LDP. When the domain sizes are all  $k$ , we have  $w = w_0^k$ .

We can then estimate the expected value of the frequency for  $Y = a$ ,

$$\Phi_{\text{GRR}}(a) = \frac{f(a)/n - q}{p - q},$$

where  $f(a)$  is the observed frequency of the perturbed value  $y_i$ .



## 2.4 CALM

Consistent Adaptive Local Marginal (CALM) [6] is an algorithm for publishing multidimensional  $k$ -way marginals via LDP. The goal is to have the curator to compute the  $k$ -way marginal from given  $\ell$ -way perturbed data such that  $k > \ell$ .

A straightforward approach is to estimate the full contingency table for  $\Omega$ . The shortcoming of this approach is that the space and time grow exponentially with the dimensionality  $k$ . Moreover, the privacy budget is linearly to the dimensionality, resulting in very noisy results.

In CALM [6], the set of users is divided into several smaller subgroups, where users perturb  $\ell$  attributes assigned to their group. The smaller  $\ell$ -way marginals, called “views,” are given as constraints for finding consistent  $k$ -way marginals using Maximum Entropy estimation via a convex optimization tool. The privacy budget can be saved because  $\ell < k$ , while accuracy decreases as the population of users  $n$  is divided into smaller groups. It is not trivial to find the optimal values for the size  $\ell$  and the number of groups. In the CALM scheme, Zhang et al. proposed an algorithm for determining a targeted threshold that would minimize estimation errors from several perspectives.

To illustrate the idea with a simple example, we reconstruct a  $k(= 3)$ -way marginal given  $\ell(= 2)$ -way views, as presented in Table 1.

Table 1: Example of views

(a) View $V_1$ ( $\ell = 2$ )	(b) View $V_2$ ( $\ell = 2$ )	(c) View $V_3$ ( $k = 3$ )																														
<table> <tr> <th></th><th><math>b_1</math></th><th><math>b_2</math></th></tr> <tr> <th><math>a_1</math></th><td>6</td><td>10</td></tr> <tr> <th><math>a_2</math></th><td>8</td><td>12</td></tr> </table>		$b_1$	$b_2$	$a_1$	6	10	$a_2$	8	12	<table> <tr> <th></th><th><math>c_1</math></th><th><math>c_2</math></th></tr> <tr> <th><math>a_1</math></th><td>4</td><td>12</td></tr> <tr> <th><math>a_2</math></th><td>6</td><td>14</td></tr> </table>		$c_1$	$c_2$	$a_1$	4	12	$a_2$	6	14	<table> <tr> <th></th><th><math>c_1</math></th><th><math>c_2</math></th></tr> <tr> <th></th><th><math>b_1</math></th><th><math>b_2</math></th></tr> <tr> <th><math>a_1</math></th><td>1</td><td>3</td></tr> <tr> <th><math>a_2</math></th><td>2</td><td>4</td></tr> </table>		$c_1$	$c_2$		$b_1$	$b_2$	$a_1$	1	3	$a_2$	2	4
	$b_1$	$b_2$																														
$a_1$	6	10																														
$a_2$	8	12																														
	$c_1$	$c_2$																														
$a_1$	4	12																														
$a_2$	6	14																														
	$c_1$	$c_2$																														
	$b_1$	$b_2$																														
$a_1$	1	3																														
$a_2$	2	4																														

Here, we have  $n = 36$  users for  $k = 3$  domains:  $\Omega_1 = \{a_1, a_2\}$ ,  $\Omega_2 = \{b_1, b_2\}$ , and  $\Omega_3 = \{c_1, c_2\}$ , with the frequencies given in the contingency table  $V_3$ , as shown in Table (1c). Two subsets of users publish their data in the corresponding contingency tables  $V_1$  and  $V_2$ , as shown in Tables (1a) and (1b), respectively.

Given the two views  $V_1$  and  $V_2$ , our goal is to find a consistent assignment for the eight variables,  $x_1, \dots, x_8$ , in  $V_3$ . Note that view  $V_1$  specifies some partial constraints related to  $\Omega_1$  and  $\Omega_2$ , as expressed in the following four simultaneous equations.

$$\begin{cases} x_1 + x_5 = 6, \\ x_2 + x_6 = 8, \\ x_3 + x_7 = 10, \\ x_4 + x_8 = 12. \end{cases} \quad (1)$$

Similarly, view  $V_2$  gives the constraints related to  $\Omega_1$  and  $\Omega_3$ , expressed as follows.

$$\begin{cases} x_1 + x_3 & = 4, \\ x_2 + x_4 & = 6, \\ x_5 + x_7 & = 12, \\ x_6 + x_8 & = 14. \end{cases} \quad (2)$$

With a privacy budget  $\epsilon$ ,  $n/2$  users in the first group perturb their data using GRR and let the curator estimate view  $V_1$ . The lefthand  $n/2$  users in the second group contribute to the estimation of view  $V_2$ . Given that the groups are exclusive, the aggregation of views requires no additional privacy budget.

The final task is to solve the equations satisfying the constraints and estimate the  $k(=3)$ -way marginals (view  $V_3$ ). This task is challenging because the views are not accurately estimated because of uncertainties, which can include noisy frequency estimation from the GRR and unbalanced group assignment. In CALM, Zhang et al. proposed the use of an off-the-shelf convex optimization tool to solve the optimization problem, expressed as

$$\begin{aligned} & \text{Maximize entropy}(x_1, \dots, x_8) \\ & \text{subject to Eq. (2) and Eq. (1)} \end{aligned}$$

### 3 Limitation of CALM

#### 3.1 Necessary number of views

It is not trivial to find the optimal size of view  $\ell$ . But, we need at least enough number of views against the size of the domains  $w_1, \dots, w_k$ . So, suppose the simple  $k$ -dimensional data where  $w_1 = \dots = w_k = w$ . In the simplest instance, we have the following relationship among  $k, \ell$  and  $w$ .

**Lemma 1** *Let  $w_1 = \dots = w_k = w$  be simple  $k$ -dimensional data. CALM with  $\ell = k - 1$  has a unique solution of  $k$ -way marginal such that  $w < k$ .*

**Proof:** With  $\ell = k - 1$ , the constraint matrix  $A$  has  $\nu = w^k$  columns and  $\mu = w^\ell \binom{k}{\ell} = w^{k-1} \binom{k}{1}$  rows. To have  $A$  nonsingular, we have  $\nu = w^k < \mu = w^{k-1}k$ , which follows  $w < k$  by dividing  $w^k$ .  $\square$

This can be extended to the following

**Lemma 2** *Let  $w_1 = \dots = w_k = w$  be simple  $k$ -dimensional data. CALM with  $\ell = k - 2$  has a unique solution of  $k$ -way marginal such that  $w^2 < k(k-1)/2$*

**Proof:** It is a straightforward from Lemma 1 by replacing  $\ell$  by  $k - 2$ .  $\square$

**Theorem 1** *Let  $w_1 = \dots = w_k = w$  be simple  $k$ -dimensional data. CALM with  $\ell$  such that  $\ell < k$  has a unique solution of  $k$ -way marginal such that*

$$w < \left( \frac{ek}{\ell} \right)^{\ell/k-\ell}$$

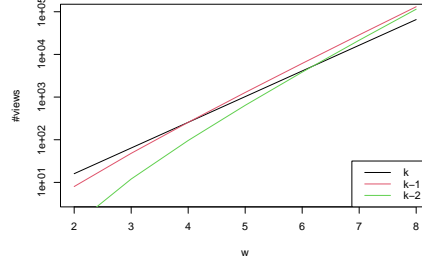


Fig. 1: The number of views for domain size  $w = |\Omega_i|$

**Proof:** Using well-known upper bound of binomial coefficient, we need a  $\nu \times \mu$  constraint matrix such that

$$\nu = w^k < \mu = w^\ell \binom{k}{\ell} < \left(\frac{ek}{\ell}\right)^\ell.$$

Taking  $k - \ell$  root after dividing  $w^\ell$  both side gives the theorem.  $\square$

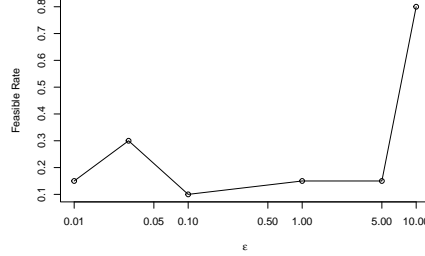
For instance, letting  $w = 4$ ,  $k = 3$ , and  $\ell = k - 1$ , we have

$$\nu = w^k = 4^3 = 64 < \mu = w^{k-1} \binom{k}{k-1} = 48,$$

which implies that the number of missing variables  $\nu$  is less than the number of equations  $\mu$ . (In CALM [6], a high-dimensional example assumed binary domains ( $w = 2$ ), for which  $\nu \geq \mu$ .) Fig. 1 shows how  $\nu$  and  $\mu$  grow with  $w$ , where  $\nu$  (shown in red) is less than  $\mu$  (black) for  $w \geq 4$ . That is, we cannot expect an accurate estimate for the  $k$ -way marginal from  $(k - 1)$ -way views. Estimation via  $(k - 2)$ -way views (green) is more robust than via  $(k - 1)$ -way views, but it is infeasible for larger domains where  $w > 6$ .

### 3.2 Infeasible CALM

CALM solves the view equations using an off-the-shelf convex optimization solver. Therefore, when the differentially privacy applies and the perturbation becomes too large, the equations may become inconsistent and the problem may become *infeasible*, meaning no solution can be found. To verify this, in Fig. 2, we show the proportion of feasible solutions when solving 3-way marginals ( $k = 3$ ) on the MovieLens dataset using CALM. As the privacy budget  $\varepsilon$  decreases, the proportion of unsolvable cases increases. For example, when  $\varepsilon = 0.01$ , only 3 out of 20 runs resulted in feasible solutions. However, the feasibility rate heavily depends on data sampling and exhibits discontinuous changes.

Fig. 2: Solvable CALM (Movie Lens,  $k = 3$ )

## 4 Proposed Method

### 4.1 Idea

The drawback of using a convex optimization tool is the stability of the solution. If we add too much large noise when perturbing private data, resulting in potentially contradictory views, it could fail to find an optimal solution, returning a “no answer” alert. If the constraints are too small, it can fail to find a unique answer and returns “undetermined”.

In this work, we propose a simple but effective way to find consistent  $k$ -way marginals without using a convex optimization tool. Our idea is to use a pseudo-inverse matrix. Before considering the details of our pseudo-inverse technique, we need to confirm that an inverse matrix will give the solution for the  $k$ -way marginals.

In our example, we first rewrite views  $V_1$  and  $V_2$  in terms of a matrix equation,

$$A\mathbf{x} = B, \quad (3)$$

where

$$A = \begin{pmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{pmatrix}$$

and  $B = (4, 6, 12, 14, 6, 8, 10, 12)^T$ . If we have the inverse of  $A$ , it is easy to solve Eq. (3) and obtain the  $k = 3$ -way marginals ( $V_3$ ) as  $\mathbf{x} = (x_1, \dots, x_8)^T$   $\hat{\mathbf{x}} = A^{-1}B$ . However,  $A$  is not always invertible.  $A$  is  $\mu \times \nu$  matrix where  $\nu$  and  $\mu$  are the number of views (contingency tables) for  $k$  and  $\ell$  dimensions, respectively. In general, it is nonsingular.

Second, we use the Moore–Penrose inverse[12], or pseudo-inverse, of matrix  $A$ . The pseudo-inverse matrix  $A^+$  of  $A$  satisfies  $AA^+A = A$ ,  $A^+AA^+ = A^+$ ,  $(AA^+)^* = AA^+$ ,  $(A^+A)^* = A^+A$ , where  $A^*$  is the conjugate transpose of  $A$ . The pseudo-inverse  $A^+$  of  $A$  is given by  $A^+ = (A^*A)^{-1}A^*$ .

A useful property of the pseudo-inverse is that it exists for any given matrix  $A$ . Therefore, without having to use a potentially unstable convex optimization, we can estimate the  $k$ -way marginals using some lower-dimensional  $\ell$ -way marginals.

Algorithm 1 shows the overall procedure.

---

**Algorithm 1** High-dimensional marginal estimation using a pseudo-inverse

---

**Require:**  $X_1, \dots, X_n \leftarrow$  input data

**Require:**  $A \leftarrow$  constraint matrix of  $\ell$ -way views

- 1: For  $\ell$ -way marginals, apply GRR perturbation to  $X_1, \dots, X_n$  and estimate  $B = (\hat{v}_1 \cdots \hat{v}_{w_\ell})^T$ , where  $\hat{v} = \Psi_{\text{GRR}}(v)$ .
  - 2: Compute the pseudo-inverse  $A^+$  of  $A$
  - 3:  $\hat{\mathbf{x}} \leftarrow A^+B$
  - 4: **return**  $k$ -way marginals  $\hat{\mathbf{x}}$
- 

## 5 Evaluation

### 5.1 Methodology

To investigate the accuracy of the proposed algorithm, we conducted an experiment using the open-data, UCI Adult dataset [13] and MovieLens datasets [14]. For three attributes, sex ( $w_1 = 2$ ), race ( $w_2 = 5$ ), and income ( $w_3 = 2$ ), GRR provided ( $\ell = 2$ )-way views. For three views, we estimated ( $k = 3$ )-way marginals using several LDP methods.

The constraint matrix  $A$  had  $\mu = 2 \times 5 + 5 \times 2 + 2 \times 2 = 24$  rows and  $\nu = 2 \times 5 \times 2 = 20$  columns. We used  $A = \begin{pmatrix} A_{1,2} \\ A_{2,3} \\ A_{1,3} \end{pmatrix}$ , where

$$A_{1,2} = (1 \ 1) \otimes \begin{pmatrix} 1 & 0 \\ & \ddots \\ 0 & 1 \end{pmatrix},$$

$$A_{2,3} = \begin{pmatrix} 1 & 0 \\ & \ddots \\ 0 & 1 \end{pmatrix} \otimes (1 \ 1) \otimes \begin{pmatrix} 1 & 0 \\ & \ddots \\ 0 & 1 \end{pmatrix},$$

$$A_{1,3} = \begin{pmatrix} 1 & 0 \\ & \ddots \\ 0 & 1 \end{pmatrix} \otimes (1 \cdots 1).$$

Table 2: Three-way marginals estimated by CALM and the proposed method

	Race	Sex	Income50k	Freq	pinv	err	CALM	err
Amer-Indian-Eskimo	Female		$\leq 50K$	107	116.6	-9.6	116.7	-9.7
Asian-Pac-Islander	Female		$\leq 50K$	303	292.1	10.9	292.3	10.7
	Black	Female	$\leq 50K$	1465	1362.4	102.7	1362.5	102.5
	Other	Female	$\leq 50K$	103	107.1	-4.1	107.2	-4.2
	White	Female	$\leq 50K$	7614	7713.8	-99.8	7713.2	-99.2
Amer-Indian-Eskimo	Male		$\leq 50K$	168	158.4	9.6	158.3	9.7
Asian-Pac-Islander	Male		$\leq 50K$	460	470.9	-10.9	470.7	-10.7
	Black	Male	$\leq 50K$	1272	1374.7	-102.7	1374.5	-102.5
	Other	Male	$\leq 50K$	143	138.9	4.1	138.8	4.2
	White	Male	$\leq 50K$	13085	12985.2	99.9	12985.8	99.2
Amer-Indian-Eskimo	Female		$> 50K$	12	2.4	9.6	2.3	9.7
Asian-Pac-Islander	Female		$> 50K$	43	53.9	-10.9	53.7	-10.7
	Black	Female	$> 50K$	90	192.7	-102.7	192.5	-102.5
	Other	Female	$> 50K$	6	1.9	4.1	1.8	4.2
	White	Female	$> 50K$	1028	928.2	99.8	928.8	99.2
Amer-Indian-Eskimo	Male		$> 50K$	24	33.6	-9.6	33.7	-9.7
Asian-Pac-Islander	Male		$> 50K$	233	222.1	10.9	222.3	10.7
	Black	Male	$> 50K$	297	194.4	102.6	194.5	102.5
	Other	Male	$> 50K$	19	23.1	-4.1	23.2	-4.2
	White	Male	$> 50K$	6089	6188.9	-99.9	6188.2	-99.2

Table 3: MAE (Movie Lens,  $\epsilon = \infty$ )

	$k = 3$		$k = 4$	
	Pinv	CALM	Pinv	CALM
mean	104.890	104.890	35.825	35.620
sd	80.140	80.140	28.976	27.229

We used CVXR [16] for a convex optimization tool for CALM and the `pracma` R library for the pseudo-inverse computations.

## 5.2 Results

Table 2 shows the ( $k = 3$ )-way marginals for the Adult dataset, where the columns headed Freq, pinv, and CALM provide the true marginals, the estimated marginals using the proposed pseudo-inverse method, and the estimated marginals using the CALM, respectively. (We did not perform any random perturbations in producing the table.)

Note that the errors for the proposed method and CALM are almost identically distributed within 0.1 precision. Mean Absolute Errors (MAEs) are 45.42 and 45.30, respectively.

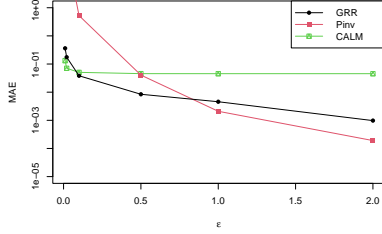


Fig. 3: MAEs with respect to privacy budget  $\epsilon$  (UCI Adult)

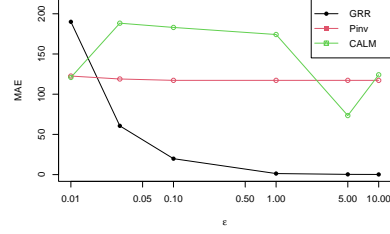


Fig. 4: MAE with respect to  $\epsilon$  (MovieLens,  $k = 3$ )

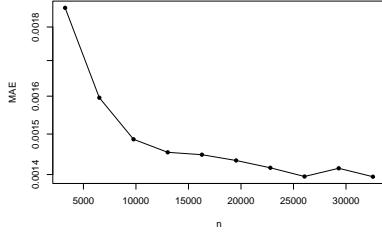


Fig. 5: MAE with respect to number of users  $n$  (UCI Adult)

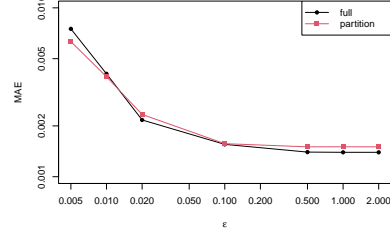


Fig. 6: MAEs for estimation from partitioned datasets (UCI Adult)

Fig. 3 shows the MAEs for the CALM, and GRR methods for a privacy budgets  $\epsilon = 0.01, \dots, 2$ . The proposed pseudo-inverse method using GRR reduces the estimation error when  $\epsilon > 1.0$ . It converges around 0.00138. We show the MAE for three LDP schemes using MovieLens dataset in Fig. 4. The MAE of GRR decreases with the increasing privacy budget as well as UCI Adult, while the MAE of CALM is unstable here. The possible reason of instability comes from the fact that any tiny inconsistency give a significant impact to the precision of CALM.

Fig. 5 shows the MAE for varying population sizes as  $n = 3256, \dots, 32561$ , which are uniformly sampled from the dataset. The MAEs represent the mean over 100 iterations. We find that MAE decreases as  $n$  increases.

Fig. 6 shows the comparison of MAEs for estimations from the full dataset (full) and when divided into three subsets (partitions). We found that both MAEs are identically distributed for all  $\epsilon$ , and the estimation from partitioned datasets sometimes gives more accurate results than using the full data ( $\epsilon < 0.02$ ).

From these results, we can conclude that the proposed pseudo-inverse estimate provides high-dimensional marginals that are as accurate as those using the conventional CALM method.

### 5.3 Discussion

We claim that the pseudo-inverse method can estimate high-dimensional marginals without using conventional optimization tools. The experiments showed small estimation errors even when no perturbation was performed. We consider that this failure comes from the singular matrix  $A$ . As we have investigated, view  $A_{1,2}$  is a  $20 \times 20$  matrix, but its rank is 16. View  $A_{2,3}$  also has a lower rank. In Section 3.1, we considered the necessary conditions for the number of views to have uniquely determined solutions. If the domain is too large, it is a challenge to find high-dimensional marginals from low-dimensional views. To improve the accuracy of the estimates, we should explore advanced strategies for estimating from lower-dimensional views.

Zhang et al. [6] studied several sources of estimation errors: noise errors, reconstruction errors, and sampling errors. Noise errors arise from perturbation processes and can be minimized by careful choice of the privacy budget according to the population of data subjects. A reconstruction error can occur when a  $k$ -way marginal is not covered by any of the chosen  $\ell$ -way marginals. This may happen for both CALM and the proposed pseudo-inverse matrix method. Sampling errors arise from biased sampling when dividing the whole population into smaller groups. Zhang et al. analyzed the variance of the marginals and considered the best approach to avoiding sampling errors.

## 6 Conclusions

We have proposed a new multi-dimensional LDP scheme that uses a pseudo-inverse matrix to estimate high-dimensional marginals from some lower-dimensional marginals. Our proposed scheme is stable, without suffering from the undetermined or absent-solution status that can occur when using convex optimization tools. Our experiment demonstrates that the proposed method estimates high-dimensional marginals as accurately as the state-of-the-art CALM method. We also explored the conditions for that the number of views to be insufficient to identify consistent marginals uniquely. Our future plans include evolving the proposed method in terms of accuracy and robustness, using a variety of sources of open-access data, and an optimal baseline LDP scheme other than GRR.

### Acknowledgment

We thank anonymous reviewers for their useful and constructive suggestions that help to improve the work significantly. This study was supported by JSPS KAKENHI Grant Number 23K11110 and JST CREST Grant Number JPMJCR21M1.

## References

1. J. Domingo-Ferrer and J. Soria-Comas, "Multidimensional randomized response," *IEEE Transactions on Knowledge and Data Engineering*, 2022, doi:10.1109/TKDE.2020.3045759.



2. X. Ren, et al., “LoPub : High-dimensional crowdsourced data publication with local differential privacy,” *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 9, pp. 2151–2166, Sept. 2018, doi:10.1109/TIFS.2018.2812146.
3. T. Wang, X. Yang, X. Ren, W. Yu, and S. Yang, “Locally private high-dimensional crowdsourced data release based on copula functions,” *IEEE Transactions on Services Computing*, p. 1, 2019.
4. X. Jiang, X. Zhou, and J. Grossklags, “Privacy-preserving high-dimensional data collection with federated generative autoencoder,” *Proceedings on Privacy Enhancing Technologies*, pp. 481–500, 2022, doi:10.2478/popets-2022-0024.
5. I. Tolstikhin, O. Bousquet, S. Gelly, and B. Scholkopf, “Wasserstein auto-encoders,” In *International Conference on Learning Representations (ICLR 2018)*, Vancouver, BC, Canada, 2018.
6. Z. Zhang, T. Wang, N. Li, S. He, and J. Chen, “CALM: Consistent adaptive local marginal for marginal release under local differential privacy,” In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security (CCS’18)*, ACM, pp. 212–229, 2018.
7. C. Dwark, F. McSherry, K. Nissim, and A. Smith, “Calibrating noise to sensitivity in private data analysis,” *TCC*, vol. 3876, pp. 265–284, 2006.
8. S. L. Warner, “Randomized response: A survey technique for eliminating evasive answer bias,” *Journal of the American Statistical Association*, pp. 63–69, 1965.
9. Ú. Erlingsson, V. Pihur, and A. Korolova, “RAPPOR: Randomized aggregatable privacy-preserving ordinal response,” In *ACM Conference on Computer and Communications Security*, pp.1054–1067, 2014.
10. “Learning with privacy at scale,” <https://machinelearning.apple.com/2017/12/06/learning-with-privacy-at-scale.html> (accessed on 2019).
11. Differential Privacy Team, “Learning with privacy at scale,” *Apple Machine Learning Journal*, vol. 1, no. 8, 2017.
12. Penrose, Roger, “A generalized inverse for matrices”, *Proceedings of the Cambridge Philosophical Society*. 51 (3), 406–413, 2008. doi:10.1017/S0305004100030401.
13. K. Bache and M. Lichman, “UCI Machine Learning Repository,” 2013, <https://archive.ics.uci.edu/ml/datasets/adult>.
14. F. Maxwell Harper and Joseph A. Konstan. “The MovieLens Datasets: History and Context,” *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5, 4, Article 19, 2015.
15. H. Kikuchi, “Castell: Scalable joint probability estimation of multidimensional data randomized with local differential privacy,” *arXiv:2212.01627*, 2022.
16. A. Fu, B. Narasimhan, and S. Boyd, “CVXR: An R package for disciplined convex optimization,” *Journal of Statistical Software*, vol. 94, no. 14, pp. 1–34, 2020, doi:10.18637/jss.v094.i14.

# Using Prior Knowledge to Improve GANs for Tabular Data Without Compromising Privacy

Sonakshi Garg<sup>1</sup>[0000-0002-7204-8228], Marcel Neunhoffer<sup>2,3</sup>[0000-0002-9137-5785], Jörg Drechsler<sup>2,3</sup>[0009-0009-5790-3394], and Vicenç Torra<sup>1</sup>[0000-0002-0368-8037]

<sup>1</sup> Umeå University, Umeå, Sweden

<sup>2</sup> Institute for Employment Research, Nuremberg, Germany

<sup>3</sup> Ludwig-Maximilians-Universität München, Germany

{sgarg, vtorra}@cs.umu.se, {marcel.neunhoffer, joerg.drechsler}@iab.de

**Abstract.** In this paper, we explore whether incorporating prior knowledge about the data can enhance GAN performance in the context of synthetic data generation for privacy protection and identify effective methodologies for doing so. We propose three approaches for integrating auxiliary information: (1) embedding public constraints into the adversarial loss function, (2) preserving correlation structures between attributes, and (3) leveraging Bayesian networks to model attribute dependencies and encode them into Conditional GANs. Through comprehensive empirical evaluations against existing baselines, we demonstrate that Bayesian networks and public constraints significantly improve the fidelity and realism of synthetic data. Furthermore, GAN-generated synthetic data lacks inherent privacy protections, making it susceptible to privacy attacks. To address this, we incorporate DP mechanisms into the GAN framework, ensuring robust privacy guarantees while maintaining data utility. The proposed approaches are evaluated for their effectiveness in generating high-quality, privacy-preserving synthetic data, offering valuable insights for future advancements in GAN-based synthetic data generation.

**Keywords:** Generative Adversarial Network · Bayesian Network · Differential Privacy.

## 1 Introduction

Generative Adversarial Networks (GANs) [16] have emerged as a groundbreaking framework in deep learning, enabling significant advancements in various domains. Introduced initially to generate high-quality synthetic data by pitting two neural networks against each other—a generator and a discriminator—GANs have proven to be remarkably successful in tasks such as high-resolution image generation [5,27], image-to-image translation [23], and even in the synthesis of continuous data distributions [35]. These capabilities have rendered GANs a cornerstone for generative modeling in computer vision and other fields.

Despite their success with continuous and high-dimensional data, GANs face considerable challenges when applied to tabular data, particularly when they contain discrete attributes. Tabular datasets, which are prevalent in domains like healthcare, finance, and social sciences, possess unique statistical characteristics. These include heterogeneous data types, highly imbalanced distributions, and intricate dependencies between variables, all of which are difficult for GANs to model effectively. Unlike image or text data with inherent structures, tabular data lacks spatial or sequential patterns and exhibits complex, high-dimensional relationships that are challenging for GANs to model. Discrete attributes further complicate generation due to their non-differentiable nature, limiting GANs' effectiveness. Several strategies have been proposed to address these challenges. Some authors [29,6] rely on a differential model by designing special functions, while [46] employ reinforcement learning to train a non-differentiable model, enabling natural language generation. Similarly, convolutional neural networks [35] and recurrent neural networks [44] have been adapted for tabular data by learning marginal distributions of columns. In addition, specialized GAN architectures, such as CTGAN [43] and CTAB-GAN [47], have been developed to tackle these issues. However, these models are constrained by their reliance on fixed assumptions about data structures and their sensitivity to training instability. While they have advanced the generation of tabular data, they still fall short in capturing the full complexity of real-world distributions [31]. A critical challenge remains in integrating domain knowledge or leveraging prior statistical information into GANs to enhance the fidelity and utility of synthetic tabular data, leaving a significant gap in achieving robust, high-quality generation.

Furthermore, a key motivation for generating synthetic data lies in its potential to serve as a privacy-preserving substitute for sensitive real-world data [11,37]. This is particularly important in contexts governed by stringent data privacy regulations such as GDPR [15] and HIPAA [19]. However, synthetic data generated by traditional GANs is vulnerable to privacy attacks [20], such as, for example, membership inference attacks [7]. Early research on private GANs focused on using Differential Privacy (DP) [12] as a privacy model by incorporating a Differentially Private Stochastic Gradient Descent (DPSGD) optimizer to update the GAN discriminator, leading to approaches like DPGAN [42]. Subsequently, alternative methods moved beyond DPGAN by introducing novel privatization techniques, often leveraging subsample-and-aggregate strategies, as seen in models like PATEGAN [25], or privately post-processing GAN samples [33]. While these approaches reported improvements over DPGAN in terms of utility, studies have highlighted their limitations in balancing utility and privacy effectively [2,31,14]. While these models introduced promising techniques, they also revealed fundamental trade-offs between data utility and privacy preservation. Achieving high-quality synthetic data that balances these trade-offs remains a significant challenge and an open research problem.

In this work, we propose methodologies to address these challenges by incorporating prior knowledge and then protecting these generators with some privacy-preserving mechanisms. We focus on **improving the quality of syn-**

**thetic tabular data.** We propose three distinct approaches to incorporate prior knowledge into GANs, enhancing their ability to generate high-quality tabular data. First, we embed public knowledge as constraints in the adversarial loss function, penalizing violations to improve the fidelity of the generated data. Second, we enforce the preservation of the original data’s correlation structure, ensuring statistical consistency. Third, we model attribute dependencies using a Bayesian network and encode these dependencies as embeddings, which are integrated into Conditional GANs (CGANs) [32] to guide the generation process. We show that these methods enable GANs to produce more realistic and statistically valid synthetic data while mitigating common issues such as mode collapse. To **ensure privacy**, we incorporate DP into our GAN framework, adapting noise injection techniques to balance privacy and utility effectively. We evaluated the proposed methodologies on multiple real-world datasets, assessing their effectiveness in improving the quality and privacy of synthetic data. Our analysis includes a quantitative evaluation of machine learning performance and correlation similarity to assess data utility, a comparative analysis with state-of-the-art GAN models. By addressing these critical gaps, our work highlights the importance of integrating prior knowledge and robust privacy mechanisms into GANs, providing a novel and practical framework for generating synthetic tabular data. This not only advances the state-of-the-art in synthetic data generation but also ensures increases privacy protection, making it highly relevant for real-world applications.

The main contributions of the paper are as follows:

- Proposal of three approaches to enhance the quality of synthetic data: using public constraints, correlation preservation, and using Bayesian networks.
- Identification of the most effective approach for generating high-fidelity synthetic data by comparing with state-of-the-art GANs.
- Utilizing a DP mechanism to ensure the privacy of the enhanced GAN-generated synthetic data.

The remainder part of the paper is organized in the following manner. Section 2 describes important concepts that have been used in the paper. Section 3 presents explanations of the proposed approach. Section 4 discusses the experimental setup and data sets involved. Section 5 provides some results and discussions. Finally, the conclusion and possible areas of future research are presented in Section 6.

## 2 Preliminaries and Related Works

In this section, we review the important concepts that are used in this paper.

### 2.1 Bayesian Network

A Bayesian Network (BN) [9] is a probabilistic graphical model representing the joint probability distribution of a set of random variables using a directed

acyclic graph (DAG). Each node corresponds to a random variable, and directed edges represent conditional dependencies. The joint probability distribution is factorized as

$$P(X_1, X_2, \dots, X_n) = \prod_{i=1}^n P(X_i \mid \text{Parents}(X_i)) \quad (1)$$

where  $\text{Parents}(X_i)$  are the parent nodes of  $X_i$ . There are several methods to learn the structure of BN from the data. Constraint-based methods (e.g., the PC Algorithm [26]) use statistical tests to identify independencies between variables. Score-based methods (e.g., Hill Climbing (HC) [40]) evaluate the quality of a network structure using a scoring criterion, such as the Bayesian Information Criterion (BIC). In this paper, we chose HC because of its efficiency in identifying probabilistic dependencies in large datasets.

The HC algorithm starts with an empty graph and iteratively adds, removes, or reverses edges between nodes. Each modification is evaluated using a scoring function like the BIC, which balances model complexity and data likelihood. The algorithm continues making improvements until it converges on the best network structure. The dependencies in the graph capture the conditional relationships between variables and serve as valuable auxiliary information. Specifically, each variable in the network has a set of parent nodes, which represent the variables that directly influence it. We used the pgmpy python package [1] to construct the BN structure as described.

## 2.2 Differential Privacy

According to the GDPR, it is crucial to ensure the privacy of personal data. To achieve this, differential privacy (DP) [12] can be employed. DP provides a formal guarantee that the inclusion or exclusion of any individual record in the dataset does not significantly affect the output, thereby protecting individual privacy.

**Definition 1.**  $(\epsilon, \delta)$ -Differential Privacy: Consider two datasets as neighboring if they differ by only one record (either by the addition or removal of a single data point). A mechanism  $A$  is said to be  $(\epsilon, \delta)$ -differentially private if, for any two neighboring datasets  $D$  and  $D'$ , and for any subset  $S$  of the output range of  $A$ , the following inequality holds:

$$P[A(D) \in S] \leq \exp(\epsilon) \times P[A(D') \in S] + \delta. \quad (2)$$

Here,  $\epsilon$  and  $\delta$  control the strength of the privacy guarantee, with smaller values providing stronger privacy. In the context of synthetic data generation, DP can be implemented in two ways: (1) by adding calibrated noise (e.g., Laplace or Gaussian noise) directly to the synthetic samples, or (2) by incorporating DP during model training, such as adding noise to gradients in the optimization process using DPSGD. This ensures that the generated synthetic data increases privacy guarantees.

### 2.3 Related Works

Over the past few years, various approaches have been proposed to improve the performance of GANs, with several focusing on incorporating prior knowledge into the model. Some recent works [41] have designed task-specific loss functions for GANs, tailoring the optimization process to improve data generation. PriorGAN [17] incorporates a Gaussian Mixture Model (GMM) prior to capturing the real data distribution, addressing issues like low-quality samples and missing modes in generated data. Feng et al. [13] introduced a method for counterfactual synthesis by studying knowledge extrapolation, allowing GANs to generate high-fidelity counterfactual results without explicit causal graph constraints. Additionally, other methods [34,27] have explored different network structures and training strategies to address issues like mode collapse and unstable training. Certain studies [18,38] adopt Bayesian principles to enhance GANs by incorporating prior distributions and posterior inference for the parameters of the generator and discriminator. StyleGAN [28] demonstrated controllable image synthesis via latent space manipulation, though it remains primarily image-focused. Subsequently, diffusion models [21] emerged, enabling conditional synthesis from class labels while offering improved stability, albeit with slower sampling rates. More recent advances include text-to-image frameworks such as Stable Diffusion v1 and transformer-based generators for structured data, exemplified by TTSGAN for time-series synthesis. Unlike prior works, our proposed method explicitly integrates auxiliary knowledge into GANs to simultaneously improve data fidelity and strengthen privacy guarantees, addressing the often-overlooked challenge of preserving realistic attribute relationships under privacy constraints.

## 3 Methodology

This paper aims to enhance the fidelity of synthetic data generated by GANs by effectively incorporating prior knowledge into the model. To achieve this, we explore and evaluate three strategies for embedding auxiliary information or imposing constraints that reflect inherent characteristics of the data. These methods are designed to guide the learning process of the GANs, ensuring the generated data aligns more closely with the underlying patterns and dependencies observed in the real dataset. Despite advancements in synthetic data generation, this problem remains unsolved, as existing methods often struggle to capture the complex relationships and prior knowledge embedded in real-world datasets, highlighting the need for more effective approaches. These techniques are described in the following subsections.

### 3.1 Public Constraint GAN (PCGAN)

Real-world data often contain inherent constraints that can be considered public information, such as logical boundaries or dependencies between variables. Incorporating these constraints into GANs can prevent the generation of implausible

or unrealistic data, thereby improving the utility and authenticity of the synthetic outputs. This approach integrates domain-specific constraints directly into the training process of the GANs by embedding them as penalty terms within the generator’s loss function. These constraints serve as additional guidance for the generator, ensuring that the synthetic data adheres to known rules or logical relationships in the real dataset. Since these constraints are assumed to be public knowledge, i.e., they apply to any data and are not dataset specific, this information can be used without concerns regarding privacy leakage. For instance, the age of humans can be constrained to lie within a realistic range of 0 to 120 years, by introducing a penalty term computed for any generated value outside this range as:

$$\text{Penalty}_{\text{age}} = \text{mean}(\max(0, -\text{age}) + \max(0, \text{age} - 120)) \quad (3)$$

These penalties are weighted and incorporated into the generator’s loss function which is defined as

$$\mathcal{L}_{\text{total}} = \lambda_{\text{adv}} \mathcal{L}_{\text{adv}} + \sum_{i \in I} \lambda_i \text{Penalty}_i \quad (4)$$

where  $I$  is the set of penalties and  $\mathcal{L}_{\text{adv}}$  is the adversarial loss and  $\lambda_{\text{adv}}$  and  $\lambda_i$ ,  $i = 1, \dots, I$  are the weights for the different loss components. During each training iteration, the generator produces synthetic samples that are evaluated against these constraints, and the computed penalty terms are back propagated along with the generator’s loss to update the generator’s parameters. This methodology ensures that the GAN generated data not only aligns with the distribution of the real dataset but also adheres to logical and practical domain-specific constraints, thereby enhancing the overall quality of the synthetic data. A detailed description of the various constraints applied, tailored to the specific datasets used, is provided in Section 4.3.

### 3.2 Correlation Structure GAN (CSGAN)

Another approach to ensure that the synthetic data closely mimics the characteristics of the original data is to align their data distributions by comparing their correlation matrices, which capture the bi-variate relationships between variables. With this method, categorical variables are first encoded using a Label Encoder [36] to enable numerical operations. The correlation matrix of the original dataset, denoted as  $C_{\text{real}}$ , is computed and used as a reference. During training, the correlation matrix of the synthetic dataset, denoted as  $C_{\text{synthetic}}$ , is also computed. Any deviation between these matrices is penalized through a custom loss function. The correlation penalty is calculated using the Frobenius norm:

$$\text{Correlation Penalty} = \|C_{\text{real}} - C_{\text{synthetic}}\|_F \quad (5)$$

where  $\|\cdot\|_F$  represents the Frobenius norm, which quantifies the element-wise differences between the two matrices. The total loss function is formulated as:

$$\mathcal{L}_{\text{total}} = \lambda_{\text{adv}} \mathcal{L}_{\text{adv}} + \lambda_{\text{corr}} \cdot \text{Correlation Penalty} \quad (6)$$

where  $\mathcal{L}_{\text{adv}}$  is the adversarial loss from the GAN training and  $\lambda_{\text{adv}}$  and  $\lambda_{\text{corr}}$  are hyper parameters controlling the trade-off between adversarial training and correlation preservation. This penalty mechanism encourages the synthetic data to maintain the variable dependencies and structural patterns inherent in the original dataset.

### 3.3 Bayesian Network GAN (BNGAN)

With this approach the objective is to effectively capture the dependencies between attributes, and utilize them as auxiliary information for the GANs. To achieve this, a BN is employed to model dependencies between attributes. BN are ideal for this problem because they explicitly model the conditional dependencies between variables, providing a structured and interpretable representation of how variables influence one another. The learned dependencies are subsequently incorporated into a Conditional GAN (CGAN) [32], serving as auxiliary information to guide the generation of realistic synthetic data. A CGAN extends the standard GAN framework by conditioning both the generator and discriminator on auxiliary information. Unlike traditional GANs, which generate data unconditionally, CGAN allows for controlled and targeted data generation by incorporating the additional input, ensuring the output aligns with the specified conditions. The proposed methodology is described in Algorithm 1, and a step-by-step explanation of the algorithm is given in the following paragraph.

The dependencies between variables are first identified using a BN as described in Section 2.1. These dependencies (parent-child relationship) are then transformed into dense vector representations (embeddings) to guide the GAN. For each parent variable, an embedding layer is initialized, where the size of the layer corresponds to the number of unique categories in that variable. The embedding layers are trained to map each categorical value to a continuous vector space, where the distances between vectors capture the semantic relationships informed by the BN structure. The embeddings of parent variables are concatenated to form a conditioning vector, which represents the combined influence of the parent variables. This conditioning vector is then passed through dense layers to generate a final representation that encapsulates the dependencies for the child variables, providing a rich latent space for generating synthetic data. This conditioning vector is finally integrated into a CGAN. Embedding layers have been widely used for learning continuous vector representations of categorical variables [30], for example, in the Word2Vec algorithm for textual data. These embeddings are trained to capture semantic relationships by mapping categorical values to a continuous vector space where distances between vectors represent the similarity between categories. We used a similar strategy to capture probabilistic dependencies between variables in BNs. This approach ensures that the synthetic data maintains the structural relationships observed in the real dataset while leveraging the flexibility of the CGAN for data generation.



---

**Algorithm 1** Bayesian Network GAN

---

**Require:**  $\mathbf{X}_{\text{train}}, \mathbf{y}_{\text{train}}$ **Ensure:**  $\mathbf{D}_{\text{syn}}$ , Performance Metrics**Step 1: Learning Variable Dependencies**

1: **Define:**  $df \leftarrow$  Dataset containing both  $\mathbf{X}_{\text{train}}$  and  $\mathbf{y}_{\text{train}}$   
2: **Sample:**  $df_{\text{sample}} \leftarrow df.\text{sample}(80\%)$   
3: **Split:**  $\text{chunks} \leftarrow \text{np.array\_split}(df_{\text{sample}}, 4)$   
4: **for** each chunk  $c \in \text{chunks}$  **in parallel do**  
5:     Initialize  $G_c \leftarrow \emptyset$   
6:     **while** no improvement in BIC score **do**  
7:          $G_c \leftarrow \text{modify}(G_c)$   
8:          $\text{score}(G_c) \leftarrow \text{BIC}(G_c)$   
9:     **end while**  
10: **end for**  
11:  $G_{\text{final}} \leftarrow \bigcup_{c=1}^4 G_c$   $\triangleright$  Union of edges from all chunks

**Step 2: Construct Dependencies and Embeddings**

12: Build dependency dictionary:  $\text{dependencies} \leftarrow$  from  $G_{\text{final}}$   
13: Identify all unique parents:  $\text{parents} \leftarrow \bigcup \text{dependencies.values}()$   
14: **for** each  $\text{parent} \in \text{parents}$  **do**  
15:     Label encode  $df[\text{parent}]$   
16:      $\mathbf{e}_{\text{parent}} \leftarrow \text{Embedding}(n_{\text{categories}}, d_{\text{embedding}})$   
17: **end for**  
18: **for** each  $\text{child} \in \text{dependencies}$  **do**  
19:     Concatenate embeddings of its parents:  $\mathbf{e}_{\text{child}} \leftarrow \big\|_{p \in \text{dependencies}[\text{child}]} \mathbf{e}_p$   
20:     Project to latent space:  $\mathbf{c}_{\text{child}} \leftarrow \text{Linear}(\mathbf{e}_{\text{child}})$   
21: **end for**  
22: Final conditioning vector:  $\mathbf{e} \leftarrow \big\|_{\text{child}} \mathbf{c}_{\text{child}}$

**Step 3: Define CGAN**

23:  $\mathcal{G} \leftarrow \text{Generator}(\mathbf{z}, \mathbf{e})$   
24:  $\mathcal{D} \leftarrow \text{Discriminator}(\mathbf{x}, \mathbf{e})$   
25:  $\mathcal{L}_{\text{adv}} \leftarrow \mathbb{E}[\log \mathcal{D}(\mathcal{G}(\mathbf{z}, \mathbf{e}))]$   
26:  $\mathcal{L}_{\text{recon}} \leftarrow \mathbb{E}[\|\mathbf{x} - \mathcal{G}(\mathbf{z}, \mathbf{e})\|^2]$

**Step 4: Train CGAN**

27: **for**  $t = 1$  **to**  $T$  **do**  
28:     Sample real data:  $\mathbf{x}_{\text{real}} \sim \mathbf{X}_{\text{train}}$   
29:     Generate synthetic data:  $\mathbf{x}_{\text{syn}} \leftarrow \mathcal{G}(\mathbf{z}, \mathbf{e})$   
30:     **Train Discriminator:**  $\mathcal{L}_D \leftarrow \mathbb{E}[\log \mathcal{D}(\mathbf{x}_{\text{real}}, \mathbf{e})] + \mathbb{E}[\log(1 - \mathcal{D}(\mathbf{x}_{\text{syn}}, \mathbf{e}))]$   
31:     **Train Generator:**  $\mathcal{L}_G \leftarrow \mathcal{L}_{\text{adv}} + \lambda \cdot \mathcal{L}_{\text{recon}}$   
32: **end for**  
33: **return**  $\mathbf{D}_{\text{syn}}$ , Evaluation Metrics

---

### 3.4 Enforcing DP for the enhanced GAN synthesizers

Obviously, standard GANs without DP guarantees will never satisfy DP. To ensure DP, a common approach is to incorporate DPSGD into the discriminator training since the GAN discriminator uses original samples to differentiate between real and fake data (the generator never sees the original data, and thus no privacy measures are required for this step). As we assume that the information used for PCGAN and CSGAN is prior knowledge, these two approaches don't use any additional information that needs to be protected and thus we can rely on this standard approach to satisfy DP. In BNGAN, which uses a Bayesian network to capture attribute dependencies and generates an embedding layer as input to the CGAN, we suggest adding Laplace noise to the embeddings to achieve DP. Given that the values of the embeddings lie within the range of  $[-1, 1]$ , the maximum possible change between two neighboring datasets is at most 2. This value serves as the global sensitivity for the Laplace mechanism, ensuring that the added noise appropriately preserves DP. Additionally, similar to the other approaches, DPSGD was applied in the discriminator training.

## 4 Experimental Setup

In this section we present the datasets used, the architecture of the GANs, and the specific constraints we enforced to modify the loss function. We provide a detailed discussion of the results of the experiments in the next section.

### 4.1 Datasets Description

In this work, we aim to incorporate prior knowledge into GANs for discrete data, particularly social science datasets rich in categorical variables. We evaluate our approach using three such datasets. The first is the Adult dataset [3], a pre-processed 1994 US Census dataset with over 45,000 individuals and attributes like education, occupation, and marital status. The second is the Social Diagnosis 2011 (SD2011) [24], a raw Polish census dataset with 35 primarily categorical attributes (e.g., education level, smoking status, work experience abroad), chosen for its real-world challenges such as missing values and outliers. The third is the German Credit Risk dataset [22], which classifies individuals as good or bad credit risks based on variables such as savings, checking amount, credit history, and credit amount. Table 1 summarizes the number of instances and attributes in each dataset. These datasets represent typical discrete social science data, where capturing inherent structure and prior knowledge is essential for effective synthetic data generation.

### 4.2 Conditional GAN (CGAN) Architecture

The Conditional GAN (CGAN) used in this paper consists of a generator and a discriminator. The generator takes a noise vector and a conditioning vector

**Table 1.** Description of Datasets

Dataset	# of Instances	# of Categorical Attr.	# of Numerical Attr.
ADULT	48842	9	6
SD2011	5000	21	14
Credit Risk	1000	6	4

as input, processes them through four dense layers with LeakyReLU activation, batch normalization (momentum = 0.8), and dropout (rate = 0.2), and outputs structured data through a final dense layer. The discriminator receives a real or synthetic sample concatenated with the same conditioning vector and processes them through four dense layers with LeakyReLU, dropout (rate = 0.4), and a final sigmoid layer for binary classification. Both components are trained using binary cross-entropy loss and the Adam optimizer (learning rate = 0.0002,  $\beta = (0.5, 0.999)$ ) for 200 epochs with batch size 32. When privacy is required, the discriminator is trained with DP-SGD using a noise multiplier of 1.1 and a max gradient norm of 1. Unlike standard GANs, this CGAN leverages a structured conditioning vector to preserve attribute relationships in discrete data.

### 4.3 Incorporating Data Constraints into the Loss Function

We enforce data constraints based on public knowledge, derived after analyzing the attributes of the dataset. For the Adult dataset, we applied an age constraint, specifying that the realistic age of a person must lie within the range [0, 120]. In this case, the penalty coefficient for the age constraint is set to 10, determined through experiments to balance adherence to realistic age ranges with maintaining data fidelity and diversity. This value ensures the generated data respects constraints without compromising quality.

For SD2011 dataset, we enforce three constraints: age constraint (similar to adult dataset), smoking constraint and work-abroad constraint. For the smoking constraint, a penalty term is computed to ensure consistency between the smoking status and the number of cigarettes smoked. Specifically, if the smoking status indicates non-smoking, the number of cigarettes smoked should be zero ( $nociga = 0$ ). The penalty for violating this constraint is defined as:

$$\text{Penalty}_{\text{smoking}} = \text{mean}((\text{smoke} < 0.5) \cdot |\text{nociga}|) \quad (7)$$

Here, *smoke* represents the smoking status (with non-smoking encoded as values below 0.5), and *nociga* represents the number of cigarettes smoked. This penalty ensures that the generated data adheres to logical dependencies between variables, enhancing its realism. For the work-abroad constraint, we enforce a penalty when the variable *workab* is "yes" (i.e., when  $\text{workab} > 0.5$ ) and the variable *wkabdur* (the duration of time worked abroad) is  $< 0$ . The penalty

is calculated using the following equation:

$$\text{Penalty}_{\text{wabroad}} = \frac{1}{n} \sum_{i=1}^n (\mathbb{I}(\text{workab}_i > 0.5) \cdot \max(0, -\text{wkabdur}_i)) \quad (8)$$

The loss function of the generator for SD2011, incorporates three constraints with a penalty coefficient of 10 for each in Eq 4. For the German Credit Risk dataset, we enforce two constraints: an age constraint and a purpose constraint. The purpose constraint applies penalties if the credit amount exceeds predefined thresholds for specific purposes, such as 5000€ for vacation or repairs and 15,000€ - 20,000€ for business or education. These thresholds were determined through an analysis of the dataset and aligned with real-world expectations, ensuring the generated data remains realistic while maintaining diversity and utility.

## 5 Results and Discussion

In this section, we empirically evaluate the generated synthetic data by assessing both, statistical properties and ML utility, and then ensure privacy by enforcing DP during data generation.

### 5.1 Impact of Synthetic Data on ML Performance

We evaluate the utility of synthetic data generated by four methods: CTGAN, PCGAN, CSGAN and BNGAN using multiple ML models. CTGAN was selected as the baseline for comparison because it is widely recognized as one of the most efficient GANs for tabular data synthesis in the literature. For classification tasks on the Adult and German credit risk datasets, we use LightGBM, XGBoostC, and Logistic Regression models, evaluating the performance based on accuracy. For the SD2011 dataset, we predict income using LightGBM regression, XGBoostR, and Linear Regression models, with performance assessed using Root Mean Squared Error (RMSE). This comprehensive evaluation ensures a robust analysis of synthetic data utility across different tasks and datasets as presented in Table 2.

Each model is trained on synthetic data and tested on real out-of-sample data. For the Adult dataset, BNGAN achieved the highest accuracy (0.78–0.79) for all ML models. For the SD2011 dataset, BNGAN also showed the lowest RMSE for all models (0.42–0.45), with PCGAN achieving comparable results (0.43–0.46). However, the SD2011 dataset contains missing values and outliers, with no pre-processing applied, leading to substantially higher error values for CTGAN (1185–1237) reflecting the challenges of working with such raw, unprocessed data. For the Credit Risk dataset, BNGAN again achieved the highest accuracy for all ML models (0.68–0.74), demonstrating the effectiveness of incorporating a BN to capture dependencies between attributes. By modeling these relationships, BNGAN generates more realistic data, improving model performance. We also note that CSGAN consistently offers the lowest utility among

all approaches for both classification tasks. We also compared the performance of synthetic data with the original data, and observed a decline in ML performance, as expected. Ideally, synthetic data should not outperform the original data since it is meant to approximate the original distribution rather than surpass it.

**Table 2.** Utility evaluations for ML models trained on synthetic data and tested on real out-of-sample data

Data	Utility Metric	ML Model	Synthetic Data				Original Data
			CTGAN	PCGAN	CSGAN	BNGAN	
ADULT	Accuracy $\uparrow$	LightGBM	0.75	0.74	0.70	<b>0.79</b>	0.87
		XGBoostC	0.75	0.73	0.69	<b>0.79</b>	0.86
		LogisticR	0.74	0.74	0.71	0.78	0.86
Credit Risk	Accuracy $\uparrow$	LightGBM	0.66	0.61	0.58	<b>0.74</b>	0.75
		XGBoostC	0.65	0.62	0.56	0.68	0.76
		LogisticR	0.67	0.63	0.59	0.70	0.74
SD2011	RMSE $\downarrow$	LightGBM	1207.35	0.44	0.48	0.43	1050.31
		XGBoostR	1236.80	0.46	0.50	0.45	1091.21
		LinearR	1185.21	0.43	0.47	<b>0.42</b>	1015.82

## 5.2 Impact of Synthetic Data on Attribute Correlation Similarity

Analyzing whether synthetic data preserve the pairwise correlations between attributes is crucial. To evaluate this, we used Cramér’s V with bias correction [4] to measure the strength of the relationship between pairs of attributes in both the original and synthetic datasets, since Cramér’s V is commonly used as a utility measure in the literature [39]. Cramér’s V is a measure of association between two categorical variables, defined as:

$$V = \sqrt{\frac{\chi^2}{n \cdot \min(k-1, r-1)}} \quad (9)$$

where  $\chi^2$  is the chi-squared statistic,  $n$  is the total number of observations,  $k$  is the number of categories in the first variable, and  $r$  is the number of categories in the second variable. The Cramér’s V values are grouped into four categories: low ( $V \in [0, 0.1)$ ), weak ( $V \in [0.1, 0.3)$ ), middle ( $V \in [0.3, 0.5)$ ), and strong ( $V \in [0.5, 1)$ ). To assess how well the synthetic data reflects the original data, we use correlation accuracy for categorical attributes, which calculates the percentage of attribute pairs where the correlation level in the synthetic data matches the original data. ’

The results in Table 3 show that the synthetic data generated by different approaches varies in preserving attribute relationships. The Adult data with its high class imbalance shows low correlation accuracy across all methods, as minority class attributes may not be well represented in the synthetic data,

causing weaker correlations between attributes. For the SD2011 dataset, PCGAN achieved the highest correlation accuracy of 0.6915, which can be attributed to the effective enforcement of domain-specific data constraints in the loss function. This allows the model to better capture the relationships between attributes. BNGAN also performed well with a correlation accuracy of 0.6780, reflecting the benefits of incorporating a BN model to capture parent-child dependencies between attributes, which helped preserve data correlations effectively. Similar trends were observed for the Credit Risk dataset, where BNGAN achieved the highest correlation accuracy of 0.6981, slightly outperforming PCGAN. Again, CSGAN showed the weakest performance in all settings.

We also measure the correlation similarity between numerical attributes by computing the Pearson correlation coefficient [8] for both real and synthetic data. This results in two correlation values:  $R_{A,B}$  for the real data and  $S_{A,B}$  for the synthetic data. The similarity between these correlation values is computed using the following formula:

$$\text{score} = 1 - \frac{|S_{A,B} - R_{A,B}|}{2} \quad (10)$$

A score of 1 indicates perfect similarity, while a score of 0 suggests no similarity. The method is adapted from SD Metrics [10], offering a standardized way to assess data quality. The results showed that BNGAN consistently achieved the highest correlation similarity, particularly for the SD2011 and Adult datasets, indicating its effectiveness in preserving numerical relationships. PCGAN also performed well, especially in the SD2011 and Credit Risk datasets, by enforcing constraints in the loss function. In contrast, CSGAN, which uses correlation-based penalties, produced lower correlation similarity scores, suggesting that it may not fully capture the complex dependency structures between attributes. Overall, both constraint-based approaches (PCGAN and BNGAN) outperform CSGAN, with BNGAN showing the strongest ability to preserve both categorical and numerical correlations across multiple datasets.

**Table 3.** Correlation Accuracy and Similarity for Categorical and Numerical Attributes

Dataset	Categorical				Numerical			
	CTGAN	PCGAN	CSGAN	BNGAN	CTGAN	PCGAN	CSGAN	BNGAN
ADULT ↑	0.3626	<b>0.4190</b>	0.3524	0.3714	0.8581	0.8843	0.8718	<b>0.8932</b>
Credit Risk ↑	0.6723	0.6812	0.6235	<b>0.6981</b>	0.8642	<b>0.8714</b>	0.8312	0.8711
SD2011 ↑	0.6684	<b>0.6915</b>	0.6123	0.6780	0.9758	0.9916	0.9468	<b>0.9971</b>

### 5.3 Results with Differential Privacy

To ensure the synthetic data generation process satisfies the definition of DP, we implemented DP a mechanism on our proposed GANs as described in 3.4

and evaluated their efficacy by comparing them with three baselines: DPGAN, PATEGAN, and ADSPAN [45]. Table 4 presents the ML performance when models are trained with DP having  $\epsilon = 1$  and  $\delta = \frac{1}{N}$ . For our DP-BNGAN model, we apply noise injection at two stages: first, during the generation of Bayesian network-based embeddings using Laplace noise with sensitivity = 2 and with  $\epsilon = 1$ , and second in the discriminator component of the CGAN with DPSGD, also with  $\epsilon = 1$ . Consequently, the total privacy budget for DP-BNGAN is  $\epsilon = 2$ . We assessed model utility across three datasets. For the Adult and Credit Risk datasets, classification accuracy was recorded using LightGBM, the best-performing model from Table 2. For the SD2011 dataset, prediction error was measured using RMSE with linear regression, also the best performing model. The results show that different models perform best on each dataset due to their ability to handle the unique characteristics of the data while preserving privacy. For Adult dataset, DP-BNGAN still performed the best, due to its strength in capturing complex distributions while maintaining privacy. PATEGAN excelled on Credit Risk dataset, due to its advanced learning capabilities, while DPGAN performs the worst on all datasets. Although the use of DP degrades the performance to some extent, the results demonstrate that the models remain highly comparable to baselines. This indicates that we are still able to preserve utility while ensuring privacy. We can also further enhance utility, at the cost of a weaker privacy guarantee.

**Table 4.** ML performance using differential privacy

Dataset	Utility Metric	DP-PCGAN	DP-CSGAN	DP-BNGAN	DPGAN	PATEGAN	ADSPAN
ADULT	Accuracy $\uparrow$	0.65	0.67	<b>0.72</b>	0.54	0.69	0.71
Credit Risk	Accuracy $\uparrow$	0.62	0.40	0.66	0.54	<b>0.96</b>	0.82
SD2011	RMSE $\downarrow$	<b>0.48</b>	0.57	0.51	0.61	0.58	0.49

## 6 Conclusion and Future Work

In this paper, we explored whether incorporating prior knowledge can enhance the performance of GANs for tabular data. We proposed three techniques for incorporating prior knowledge into GANs without compromising the privacy of personal data. These methods aim to provide better control over the output of the GAN. Our comparative analysis with baseline models revealed that using a Bayesian network to capture attribute dependencies significantly improved data quality, as validated through various ML and statistical evaluations. Additionally, enforcing public knowledge as constraints also enhanced performance in certain cases, making both approaches viable for future applications. To ensure privacy, we integrated a DP mechanism into the GAN training process. In future work, we aim to extend this comparison to a broader range of privacy-preserving GANs to further highlight our contributions relative to existing methods. Also,

we plan to explore other probabilistic models to further improve the quality of synthetic data and investigate more novel ways of incorporating prior knowledge into GANs.

## References

1. Ankan, A., Panda, A.: pgmpy: Probabilistic graphical models using python. In: Proceedings of the Python in Science Conference. SciPy, SciPy (2015). <https://doi.org/10.25080/majora-7b98e3ed-001>, <http://dx.doi.org/10.25080/Majora-7b98e3ed-001>
2. Arnold, C., Neunhoeffler, M.: Really useful synthetic data—a framework to evaluate the quality of differentially private synthetic data. arXiv preprint arXiv:2004.07740 (2020)
3. Becker, B., Kohavi, R.: Adult. UCI Machine Learning Repository (1996), DOI: <https://doi.org/10.24432/C5XW20>
4. Bergsma, W.: A bias-correction for cramér’s v and tschuprow’s t. Journal of the Korean Statistical Society **42**(3), 323–328 (2013)
5. Brock, A.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018)
6. Che, T., Li, Y., Zhang, R., Hjelm, R.D., Li, W., Song, Y., Bengio, Y.: Maximum-likelihood augmented discrete generative adversarial networks. arXiv preprint arXiv:1702.07983 (2017)
7. Chen, D., Yu, N., Zhang, Y., Fritz, M.: Gan-leaks: A taxonomy of membership inference attacks against generative models. In: Proceedings of the 2020 ACM SIGSAC conference on computer and communications security. pp. 343–362 (2020)
8. Cohen, I., Huang, Y., Chen, J., Benesty, J., Benesty, J., Chen, J., Huang, Y., Cohen, I.: Pearson correlation coefficient. Noise reduction in speech processing pp. 1–4 (2009)
9. Cooper, G.F., Herskovits, E.: A bayesian method for the induction of probabilistic networks from data. Machine learning **9**, 309–347 (1992)
10. DataCebo, Inc.: Synthetic Data Metrics (2023), <https://docs.sdv.dev/sdmetrics/>
11. Drechsler, J.: Synthetic datasets for statistical disclosure control: theory and implementation, vol. 201. Springer Science & Business Media (2011)
12. Dwork, C.: Differential privacy. In: International colloquium on automata, languages, and programming. pp. 1–12. Springer (2006)
13. Feng, R., Xiao, J., Zheng, K., Zhao, D., Zhou, J., Sun, Q., Zha, Z.J.: Principled knowledge extrapolation with gans. In: International Conference on Machine Learning. pp. 6447–6464. PMLR (2022)
14. Fössing, E., Drechsler, J.: An evaluation of synthetic data generators implemented in the python library synthcity. In: International Conference on Privacy in Statistical Databases. pp. 178–193. Springer (2024)
15. GDPR-EU: Gdpr compliance. <https://gdpr.eu/>
16. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial networks. Communications of the ACM **63**(11), 139–144 (2020)
17. Gu, S., Bao, J., Chen, D., Wen, F.: Priorgan: Real data prior for generative adversarial nets. arXiv preprint arXiv:2006.16990 (2020)



18. He, H., Wang, H., Lee, G.H., Tian, Y.: Bayesian modelling and monte carlo inference for gan. In: International Conference on Learning Representations. vol. 3, p. 4 (2019)
19. HIPAA-US: Hipaa compliance. <https://www.hhs.gov/hipaa/for-professionals/privacy/>
20. Hitaj, B., Ateniese, G., Perez-Cruz, F.: Deep models under the gan: information leakage from collaborative deep learning. In: Proceedings of the 2017 ACM SIGSAC conference on computer and communications security. pp. 603–618 (2017)
21. Ho, J., Jain, A., Abbeel, P.: Denoising diffusion probabilistic models. *Advances in neural information processing systems* **33**, 6840–6851 (2020)
22. Hofmann, H.: Statlog (German Credit Data) (1994), DOI: <https://doi.org/10.24432/C5NC77>
23. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017)
24. J., C., T., P.: SD2011. <http://www.diagnoza.com/index-en.html> (2011)
25. Jordon, J., Yoon, J., Van Der Schaar, M.: Pate-gan: Generating synthetic data with differential privacy guarantees. In: International conference on learning representations (2018)
26. Kalisch, M., Bühlman, P.: Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *Journal of Machine Learning Research* **8**(3) (2007)
27. Karras, T.: Progressive growing of gans for improved quality, stability, and variation. *arXiv preprint arXiv:1710.10196* (2017)
28. Karras, T., Laine, S., Aila, T.: A style-based generator architecture for generative adversarial networks. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 4401–4410 (2019)
29. Kusner, M.J., Hernández-Lobato, J.M.: Gans for sequences of discrete elements with the gumbel-softmax distribution. *arXiv preprint arXiv:1611.04051* (2016)
30. Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J.: Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems* **26** (2013)
31. Miletic, M., Sariyar, M.: Challenges of using synthetic data generation methods for tabular microdata. *Applied Sciences* **14**(14), 5975 (2024)
32. Mirza, M.: Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784* (2014)
33. Neunhoffer, M., Wu, S., Dwork, C.: Private post-gan boosting. In: International Conference on Learning Representations (2021), <https://openreview.net/forum?id=6isfr3JCbi>
34. Parimala, K., Channappayya, S.: Quality aware generative adversarial networks. *Advances in neural information processing systems* **32** (2019)
35. Park, N., Mohammadi, M., Gorde, K., Jajodia, S., Park, H., Kim, Y.: Data synthesis based on generative adversarial networks. *arXiv preprint arXiv:1806.03384* (2018)
36. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in python. *Journal of machine learning research* **12**(Oct), 2825–2830 (2011)
37. Rubin, D.B.: Statistical disclosure limitation. *Journal of official Statistics* **9**(2), 461–468 (1993)
38. Saatci, Y., Wilson, A.G.: Bayesian gan. *Advances in neural information processing systems* **30** (2017)

- 
39. Tao, Y., McKenna, R., Hay, M., Machanavajjhala, A., Miklau, G.: Benchmarking differentially private synthetic data generation algorithms. *arXiv preprint arXiv:2112.09238* (2021)
  40. Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max-min hill-climbing bayesian network structure learning algorithm. *Machine learning* **65**, 31–78 (2006)
  41. Wu, J.L., Kashinath, K., Albert, A., Chirila, D., Xiao, H., et al.: Enforcing statistical constraints in generative adversarial networks for modeling chaotic dynamical systems. *Journal of Computational Physics* **406**, 109209 (2020)
  42. Xie, L., Lin, K., Wang, S., Wang, F., Zhou, J.: Differentially private generative adversarial network. *arXiv preprint arXiv:1802.06739* (2018)
  43. Xu, L., Skoularidou, M., Cuesta-Infante, A., Veeramachaneni, K.: Modeling tabular data using conditional gan. *Advances in neural information processing systems* **32** (2019)
  44. Xu, L., Veeramachaneni, K.: Synthesizing tabular data using generative adversarial networks. *arXiv preprint arXiv:1811.11264* (2018)
  45. Yoon, J., Drumright, L.N., Van Der Schaar, M.: Anonymization through data synthesis using generative adversarial networks (ads-gan). *IEEE journal of biomedical and health informatics* **24**(8), 2378–2388 (2020)
  46. Yu, L., Zhang, W., Wang, J., Yu, Y.S.: Sequence generative adversarial nets with policy gradient. 492 in. In: *AAAI conference on artificial intelligence*. vol. 493 (2017)
  47. Zhao, Z., Kunar, A., Birke, R., Chen, L.Y.: Ctab-gan: Effective table data synthesizing. In: *Asian Conference on Machine Learning*. pp. 97–112. PMLR (2021)

## **Session 3: Applications**

# Lessons from a Robotaxi: Challenges in Selecting Privacy-Enhancing Technologies

Ala'a Al-Momani<sup>1</sup>[0000–0001–5752–7338], David Balenson<sup>2</sup>[0000–0002–0913–4852],  
Christoph Bösch<sup>3</sup>[0000–0001–9312–8000], Zoltán Ádám Mann<sup>4</sup>[0000–0001–5741–2709],  
Sebastian Pape<sup>5,6</sup>[0000–0002–0893–7856], and Jonathan Petit<sup>7</sup>[0000–0002–8644–1442]

<sup>1</sup> Ulm University

<sup>2</sup> USC Information Sciences Institute

<sup>3</sup> Bosch Research

<sup>4</sup> University of Halle-Wittenberg

<sup>5</sup> Continental Automotive Technologies GmbH

<sup>6</sup> Goethe University Frankfurt

<sup>7</sup> Qualcomm Technologies Inc.

**Abstract.** Engineering privacy-friendly systems requires first assessing privacy threats and then selecting privacy-enhancing technologies (PETs) to mitigate the threats. While well-established methods such as LINDDUN support threat assessment, systematic approaches for PET selection remain underdeveloped. This paper presents our experience applying three such approaches to a realistic robotaxi use case. Although each method has been validated by its respective authors on simple use cases, we found that none could adequately support PET selection in our complex, real-world scenario. As a result, we also explored a pragmatic approach based on Hoepman’s privacy strategies. By analyzing the strengths and limitations of these approaches, we identify key challenges that PET selection methodologies should address and provide recommendations to guide the future development of such methodologies.

**Keywords:** privacy-enhancing technologies · PET selection · privacy threats · privacy threat mitigation · privacy engineering · robotaxi.

## 1 Introduction

For the early phases of the privacy engineering process — such as privacy threat assessment — several methodologies provide specific guidance (e.g., LINDDUN [28], PANOPTIC [18], and xCOMPASS [9]). These methodologies support the high-level design of privacy-friendly systems reasonably well, often through the use of privacy strategies and privacy patterns [13]. Academic efforts have also proposed ways to support later phases, in particular the selection of Privacy-Enhancing Technologies (PETs) to address the found privacy threats. Such work draws on privacy principles [24], best practices, activities, objectives, patterns [17, 25], strategies [13], and threat models [8], as well as the broader concept of privacy

by design [11]. However, the practical applicability of these proposals is not fully understood. Applying them to the detailed design of privacy-friendly systems in the real world may be challenging because of the approaches' high level of abstraction and other limitations and shortcomings.

This work investigates how the PET selection problem can be solved in practice, using a realistic robotaxi system as use case. Robotaxi services involve extensive and sensitive data processing throughout their lifecycle — from ride requests and routing to post-ride analytics — making them an ideal testbed for evaluating PET selection methodologies. Our aim is to investigate to what extent existing methodologies can be used to select appropriate PETs to enhance the privacy in the considered robotaxi service. In this work, we do not propose the final design of a privacy-preserving robotaxi service, but rather focus on investigating the methodologies for selecting PETs.

We make the following contributions: i) We identify three methodologies in the literature that promise guidance on PET selection, and apply them to a realistic robotaxi use case. We find that none yield satisfactory results. ii) We apply a pragmatic, experience-based approach based on Hoepman's privacy strategies [13] to identify a useful set of PETs. iii) We analyze the strengths and limitations of these approaches and extract insights to inform the development of improved PET selection methodologies. Our findings show that existing methodologies provide limited — or no — support for the detailed design and actual implementation of privacy-friendly systems. In particular, there is a lack of systematic, actionable support for selecting PETs as well as clear guidance how to implement and configure the selected PETs, how to combine them effectively, and how to integrate them into an overall system.

## 2 Related Work

We identified several privacy frameworks and projects. They cover the areas of privacy engineering (STRAP [15], which builds on prior work by Bellotti and Sellen [6] and Hong et al. [14]), system re-engineering (POSD [5]), privacy by design (PRIPARE<sup>8</sup> based on the work of Kung [19] and Hoepman [13]), and compliance (PARROT [4]). MITRE has released the Privacy Engineering Framework and Life Cycle Adaptation Guide<sup>9</sup>, while ENISA has published the PETs Control Matrix<sup>10</sup> and a report on data protection engineering<sup>11</sup>. However, none of these frameworks give specific support in the selection of PETs.

Several relevant standards also exist. ISO/IEC 27701 extends ISO/IEC 27001 by adding requirements for establishing and improving a Privacy Information Management System (PIMS). ISO/IEC 27550 describes privacy engineering across the system lifecycle, drawing from Hoepman's privacy strategies [13] and Privacy

<sup>8</sup> <https://pripareproject.eu/>

<sup>9</sup> <https://www.mitre.org/sites/default/files/2021-11/>

<sup>10</sup> <https://www.enisa.europa.eu/news/enisa-news/enisas-pets-control-matrix-a-tool-to-evaluate-online-and-mobile-privacy-tools>

<sup>11</sup> <https://www.enisa.europa.eu/publications/>

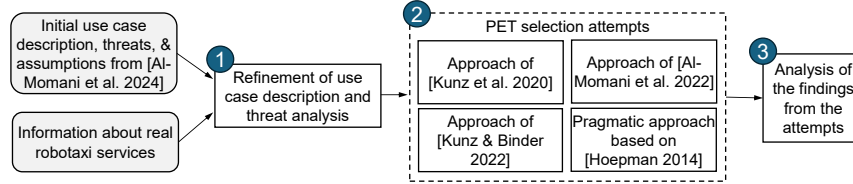


Fig. 1: Overview of the methodology used in this paper

Control Examples that are similar to patterns (e.g., Hide: Encryption, Mixing, Perturbation). Similar to NIST SP800-53, ISO/IEC 29151 defines objectives, controls, and guidelines for implementing controls for protecting personally identifiable information (PII). Yet, none of these standards provide specific support for selecting PETs.

In the academic literature, Drozd and Dürmuth [10] suggested linking privacy patterns to PETs, but only as a conceptual outlook. Pape et al. [24] proposed selecting PETs based on GDPR principles, without referencing specific threats. Adams [1] introduced a privacy tree to classify PETs, offering some guidance for selection, but the list is incomplete and several leaves are linked to multiple PETs. Jordan et al. [16] provide an extensive list of PETs, but offer minimal support for selecting. We only found three papers that provide specific guidance in PET selection [3, 20, 21], which we discuss in greater detail in Section 5.

As our use case is in the automotive domain, we also examined PET-related literature in this area. Al-Momani et al. [2] explored the usefulness of privacy patterns in improving privacy in future automotive systems. Chah et al. [7] applied LINDDUN to analyze privacy threats. Pape et al. [26] proposed a system model to identify suitable integration points for PETs in a vehicle. Löbner et al. [22] evaluated de-identification techniques in automotive use cases. None of these works proposed a methodology for selecting suitable PETs.

### 3 Methodology

Fig. 1 depicts the methodology used to perform the research reported in this paper. Our methodology is structured around a *refined robotaxi use case* derived from Al-Momani et al. [2]. We enhanced this use case to reflect more realistic data flows and service phases based on descriptions from real providers like Waymo and Uber<sup>12</sup>. We carefully checked that these refinements did not alter the original threat model or its underlying assumptions. As a result, we were able to reuse the *threat assessment* conducted by Al-Momani et al.[2].

To *identify suitable PETs* for our use case, we applied three PET selection approaches from the literature: i) Kunz et al. [20] who propose a reproducible method for selecting data-dependent PETs that can be used independently or

<sup>12</sup> cf. <https://waymo.com> and <https://www.uber.com>, respectively

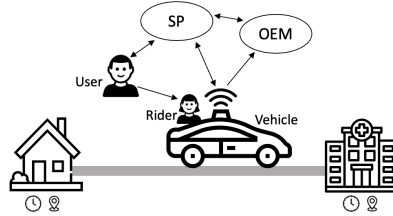


Fig. 2: Basic system model of a robotaxi service, from [2].

alongside other methods; ii) Kunz and Binder [21] who offer an application-oriented classification of PETs based on privacy protection goals, functional context, technology maturity, and impact on various non-functional requirements; and iii) Al-Momani et al. [3] who employ decision trees to guide the selection of privacy solutions based on LINDDUN threats and Hoepman's privacy strategies [13]. In addition to these approaches, we applied a *pragmatic, experience-driven approach* (cf. Sect. 5.4) in which we revisited assumptions, analyzed the purpose of data processing, and considered applicable PETs. We then *analyzed the outcomes* to uncover key challenges, limitations, and differences across the approaches. All steps and findings were collaboratively reviewed to ensure consistency.

#### 4 Use Case: Robotaxi – Refined System Model

Robotaxi services, which are autonomous, driverless taxi systems, represent a cutting-edge application of self-driving vehicle technology. By focusing on a generic robotaxi service, our aim is to derive insights applicable across the broader industry, rather than to a single provider. From a privacy perspective, a robotaxi service differs significantly from a traditional taxi service. In a traditional taxi, the driver handles not only the driving, but also rider interaction, payment, and unexpected situations. In a robotaxi, these functions are performed by a combination of artificial intelligence and a remote service provider. As a result, more data may need to be collected to ensure safe and effective service operation.

Our system model builds on the robotaxi model proposed by Al-Momani et al. [2], providing a refined system version that offers closer alignment with real-world deployments. This refinement is based on examining existing services and incorporates best practices from the industry. While it does not (intentionally) address privacy enhancements, the refined model serves as a more practical foundation for selecting applicable PETs to mitigate the identified privacy threats.

Additionally, we noticed during the application of the pragmatic approach that all of the three investigated approaches require a clean use case description with minimal assumptions. Therefore, we revisited the original assumptions, asking if the data in question was truly necessary and if it could be reduced. For instance, we challenged the assumption that a user's birth date needs to be collected during registration, as a more privacy-friendly option would be to use





**3. Vehicle Assignment & Ride Confirmation.** The system assigns an autonomous vehicle and provides ride details to *User*. **Data Collected:** Vehicle identification (e.g., make, model, license plate), estimated time of arrival (ETA), and *Rider*'s updated location for precise pickup (if selected). **Purpose:** To inform *User* of vehicle details and ensure accurate pickup coordination. **Additional Features:** *User* is notified when the vehicle arrives. Identity confirmation (e.g., PIN) is required to ensure the correct *Rider* enters the vehicle. Additionally, the vehicle assignment requires fleet management data, including the precise location of vehicles and the current fuel or battery levels.

**4. Ride Execution.** The autonomous vehicle navigates to the destination, guided by its sensors and real-time data processing. **Data Collected:** Real-time vehicle location, internal and external sensor data (e.g., audio, cameras, LIDAR) and user interaction data within the vehicle (e.g., temperature or music preferences). Sensor data, camera data, and vehicle location are also accessible to the *OEM* at any time. **Purpose:** To enable safe travel, ensure *Rider* comfort, and provide operational support. **Additional Features:** *Rider* may change the route or drop-off location and can contact customer support via vehicle interface or the app if issues arise.

**5. Payment and Feedback.** Payment is processed automatically upon ride completion. *Rider* can provide feedback via the vehicle interface, and *User* via the application. **Data Collected:** Ride fare details, payment method, trip history, and user feedback (e.g., ratings, comments). **Purpose:** To complete the financial transaction, maintain a record of rides, and improve service quality based on feedback. **Additional Features:** *User* may receive trip summaries, and promotional offers or discounts are applied based on *User*'s profile.

**6. Post-Ride Actions.** Additional interactions may occur between *User* and *SP*, including invoice creation, ride history and analytics, customer support, loyalty programs and rewards, safety and security issues, service customization, data deletion, subscription cancellation, and social media sharing. **Data Use:** Depending on the action, different existing data items may be reused or new data may be collected.

## 5 PET Selection

Al-Momani et al. [2] conducted a privacy threat assessment of the original use case. Because our refined use case closely aligns with the original, particularly in terms of privacy threats, the assessment remains applicable, and we refer readers to the original paper for more details. Our current focus is on selecting PETs to mitigate these threats.

Our literature review identified three approaches that offer specific guidance for PET selection. In Sections 5.1-5.3, we describe our experience applying these methods to the robotaxi use case. Given the limitations we encountered, we also applied a pragmatic approach based on Hoepman's privacy strategies [13]. The challenges reported in Sections 5.1-5.4 are not intended as criticisms of these approaches. We recognize these approaches are valuable initial steps toward

addressing a complex problem. Our goal is to highlight that the current state of the art in PET selection remains inadequate for handling realistic use cases.

### 5.1 Approach of Kunz et al. (2020)

Kunz et al. [20] proposed a methodology for selecting PETs for IoT-based services, with a focus on the automotive domain. The methodology consists of four steps: service description, data-driven elicitation, service-driven elicitation, and PET selection. We go through these four steps and try to apply them to our use case.

**A. Service description.** In this step, the service is specified, focusing on the required data and the purposes of data processing. We have done this in Sect. 4.

**B. Data-driven elicitation.** In this step, all data identified in the first step is analyzed according to 6 criteria: continuous or categorical data, set size, ordinal or nominal data, data longevity, value sequences, metadata and identifiers. Each of these analysis steps should help narrow down the set of PETs applicable to the given type of data. In our case, this requires quite some effort. We identified 29 data types in our use case (see Table 1), leading to  $29 \cdot 6 = 174$  analysis steps. We present here only a couple of those steps as examples.

One criterion is whether the data is continuous or categorical, which poses a challenge since most of our data types (e.g., name, address, vehicle ID, route) are neither continuous nor categorical. Some data (e.g., fare) is continuous. The analysis tells us that some PETs, for example PRAM (post-randomization method), cannot be applied to these data types. Similarly, some of our data (e.g., payment method) is categorical, and the analysis tells us that some PETs, for example noise masking, cannot be applied to these data types. Another criterion is the number of values that the given data type can assume. For most of our data types, this depends on implementation details (e.g., the string length maximally allowed for name or address). This seems to contradict the statement of Kunz et al. that their methodology can be applied in the early phases of the system design process, because such choices may not have been made yet at this stage. Also, Kunz et al. do not specify what to do with this information. They only state that a smaller set of possible values decreases the applicability of PETs. It is not clear how this could help narrow down the set of applicable PETs.

**C. Service-driven elicitation.** This step entails analyzing the service’s requirements on data utility, with the aim of determining which PETs would not undermine the usefulness of the given service. For this purpose, the methodology uses three criteria: value precision, data freshness, and attribute dependency.

As to the first criterion, the “precision required by the service” is unclear for certain data types (e.g., camera feed). For other data types, the precision requirement may vary over time: e.g., the pick-up location must be known exactly when the vehicle picks up the rider, but the precision may be lowered when this data is stored for later processing. Unfortunately, the methodology does not support such varying precision requirements. The second criterion is how fresh the data needs to be. This is again problematic: the same data can be associated with different freshness requirements for different purposes. For example, if the robotaxi encounters a difficult traffic situation and requires remote control from a

human operator, that operator needs the camera feed in real time. On the other hand, for settling compensation claims, there may be a need to access archived camera feeds from weeks before. Again, the methodology does not support this type of varying requirements. The last criterion is the dependency between attributes. Indeed, some of the data types in our use case are not independent. For example, there is a connection between the route and the fare, since a longer route typically leads to a higher fare. Kunz et al. draw our attention to the fact that in such cases, determining different PETs for the dependent attributes may cause problems. It is not clear how this information could help our PET selection process, since the different data types may force us to use different PETs for those attributes. Also, even if the same PET is used for two interdependent attributes, the dependency may still cause problems if not properly taken into account, and the methodology does not clarify how to avoid such problems.

**D. PET selection.** Assuming that the previous two steps delivered a set of potentially applicable and useful PETs (which is not the case in our use case due to the difficulties reported above), this step aims at choosing the best ones from those sets. Unfortunately, Kunz et al. state that this is highly use-case-specific, so that they do not provide a systematic approach for this step.

**Further limitations.** As we saw above, steps B and C are only partially applicable to our use case, and step D does not give clear guidance. In addition, the approach suffers from further limitations. First, the approach is limited to data-obfuscation PETs. In our case, several data types (e.g., user name or payment information) must be available to the service provider without modifications for legitimate purposes, so that they cannot be obfuscated. There are data protection requirements associated with these data types, but addressing these requirements requires PETs not supported by the methodology. Second, the approach assumes a list of available PETs. However, finding the right level of abstraction for PETs is challenging. E.g., Kunz et al. consider aggregation to be one PET, but mention that various aggregation techniques exist. Those techniques could be just as well considered individual PETs. If we find out using the methodology that we should use aggregation, we are still faced with the question of which aggregation technique to use. Third, Kunz et al. state that their approach can be used in tandem with LINDDUN. However, the approach excludes two important threats covered by LINDDUN: unawareness and non-compliance. Compliance with data protection regulations is the primary privacy objective for most service providers, making non-compliance the most important threat from their point of view.

## 5.2 Approach of Kunz and Binder (2022)

Kunz and Binder [21] propose a categorization of PETs to aid PET selection. For each considered PET, they determine the relevant privacy goals, metrics for measuring the PET's privacy effect, the relevant "functional scenario" (one of: release, messaging, authentication, authorization, retrieval, computation), the PET's maturity on a scale from 1 to 3, and the PET's impact on performance, architecture, and utility (the last three are binary attributes: there is either impact or not). The paper provides this categorization for 29 PETs. On this

basis, the following methodology can be deduced. Starting from a privacy threat assessment, first the privacy goal and functional scenario is determined for each threat. Then, the categorization helps identify the subset of PETs applicable to the combination of privacy goal and functional scenario. Finally, the maturity and impact attributes of the short-listed PETs help choose the most appropriate PET. In the following, we go through these steps, applying them to our use case.

**A. Identifying privacy goal and functional scenario.** A privacy threat assessment of our use case has already been performed by Al-Momani et al. [2] using LINDDUN. The privacy goals used by Kunz and Binder are directly linked to the LINDDUN threat types, which makes it trivial to determine the privacy goal related to each threat. E.g., for a linkability threat, the related privacy goal is unlinkability. Determining the “functional scenario” that provides the context for a threat, however, is not always obvious. Some threats arise in the context of activities that could belong to more than one category: e.g., the threats arising from data sharing between the *SP* and the *OEM* could be seen to belong to both the “release” and the “messaging” category. The functional scenario of some other threats—e.g., the threat of storing personal data beyond its necessary retention period—does not seem to belong to any of the proposed categories.

**B. Identifying relevant subset of PETs.** If the privacy goal and the functional scenario could be determined for a threat, then the matrix of Kunz and Binder can be used to mechanically determine the subset of relevant PETs. Even this seemingly straightforward step poses difficulties. The matrix offers no PETs for unawareness and non-compliance threats, although, as we mentioned earlier, these threats can be very important. Also, there are many combinations of privacy goal and functional scenario, for which the matrix offers no PETs.

**C. Selecting the most appropriate PET.** If we managed to identify a set of applicable PETs for a given threat through the two previous steps, then the final step is to select the most appropriate one. Unfortunately, the paper offers no clear guidance on how to do that. It is suggested that the maturity and the impact on performance, architecture, and utility should be helpful in making this decision. But it is not clear how. E.g., suppression and recoding are given as two PETs that can both address linkability threats in a “release” functional scenario, and they have the same maturity and the same impact on performance, architecture, and utility, so it remains unclear which one to choose. Another example: swapping and noise masking can be used for the same type of threat and functional scenario; swapping has a lower maturity than noise masking, but noise masking impacts utility, making it unclear which one to choose.

**Further limitations.** Beyond the questions that the individual steps raise, the approach also suffers from more general issues. Some are similar to the problems identified in Sect. 5.1. E.g., unawareness and non-compliance are missing in both approaches. Also, we mentioned in Sect. 5.1 that it is difficult to come up with a good list of PETs because it is not clear if different variants of a PET should be regarded as different PETs. For the method of Kunz and Binder, this problem is even more severe because different variants of a PET may have different maturity and different impact on performance, architecture, and utility. E.g., Kunz and

Binder mention synthetic data as a PET. However, there are many ways to generate synthetic data, and their impact on, e.g., utility can be very different.

The impact attributes of Kunz and Binder are problematic anyway. It is not possible to capture the impact of a PET on performance, architecture, and utility in general, because this depends on many further details. E.g., the matrix of Kunz and Binder shows that the PET MPC (multi-party computation) impacts performance. However, there are many MPC techniques, and their performance impact is very different. Even for one particular MPC technique, e.g., additive secret-sharing, its performance impact depends heavily on the types of operations that it is applied to: linear operations (addition or multiplication by a constant) can be very quickly performed on additively secret-shared numbers, whereas non-linear operations are much more costly [27]. Thus, the performance impact depends not only on the PET, but also on the context in which it is applied. A further problem is that the analysis must be performed for every single threat. In a real system, the number of threats can be high, making this impractical. Also, the risk posed by several threats may simply be accepted or may be addressed by non-technical means, so that PET selection for these threats is not necessary. E.g., in our use case, there are obvious identifiability threats stemming from the collected identifiers, but this is accepted because of other requirements. Finally, threats may be connected to each other. The methodology proposes a PET for each threat independently, potentially leading to a sub-optimal solution.

### 5.3 Approach of Al-Momani et al. (2022)

Al-Momani et al. [3] propose a methodology using decision trees to systematically guide users from privacy threats identified with LINDDUN to suitable privacy solutions. For this, specific key nodes are identified in the LINDDUN threat trees. These nodes contain information regarding the cause of the threat, the threat class, and the system element where the threat applies. For each key node, the mitigation goal is defined, and nodes sharing the same goal are grouped together. In total, ten mitigation goals are defined. For each mitigation goal, potential countermeasures are defined and then ordered according to the data-oriented privacy design strategies [13], i.e., Minimize, Separate, Abstract, and Hide. This process yielded four solution trees for the mitigation goals “protect-attributes”, “protect-communication-metadata”, “protect-id”, and “secure-processing”. In the following, we apply this approach to our use case.

**A. Identify “key nodes” for the solution trees.** To select the applicable PETs, the original approach had to be modified because it had been designed for an earlier version of LINDDUN, rendering the utilization of the key nodes unfeasible. Our adaption process was initiated by mapping the identified threats from the LINDDUN analysis to the solution trees. To maintain a fundamental element of the method—the usage of the rationales underlying a threat identified through the threat trees—we used the assumptions from the use case [2], which encompass analogous information and facilitated the mapping process.

**B. Identify possible PETs using the solution trees.** The aforementioned new mapping allowed us to use the solution trees, which consequently resulted in

some PETs for the different phases. The first step is to address the applicability of a PET. Then, it is necessary to determine whether the PET alone is adequate to remedy the threat of the key node or if it must be combined with other applicable PETs. In summary, we observed two main outcomes of the method per threat: i) Mitigation is not applicable since the (precise) data is required for the service, e. g. for user identification; and ii) Mitigation is possible using: Remove, Replace, Separate, or use Noisy & less granular attributes, depending on the data.

The proposed solution trees are a promising concept, particularly in terms of prioritizing privacy strategies and assessing the necessity of data. This approach involves determining whether the data is indispensable and, if so, explores options for its replacement, separation, or generalization. Only after this thorough evaluation should the utilization of advanced PETs be considered. However, this method also has major shortcomings. The “*secure-processing*” tree might be complete regarding PETs, since it helps choose one of the three currently available PETs for secure processing: homomorphic encryption, trusted execution environments, and multiparty computation. However, the “*protect-id*” tree considers only attribute-based credentials as a PET which limits usability. The “*protect-attributes*” tree only considers encryption in general and no specific PET. Although the key ‘entry’ nodes include “Untrusted communication”, “Observe message and/or channel”, and “Dataflow not fully protected”, even TLS is missing as a PET. In addition, technologies that protect attributes are missing, such as attribute-based credentials or zero-knowledge proofs. The “*protect-communication-metadata*” deals with “Non-anonymous Communication” and lists only Onion routing and Hiding timestamps and the message size by random padding as possible PETs.

**Further Limitations.** The approach suggests primarily to use Hoepman’s privacy strategies [13], but lacks more concrete details on PET selection. Missing PETs limit the selection of (advanced) technical PETs.

#### 5.4 A Pragmatic Approach Based on Hoepman (2014)

We now sketch a pragmatic approach based on Hoepman’s privacy design strategies [13] and the authors’ collective expertise. Al-Momani et al. [2] previously identified the assumptions underlying the privacy threats they found. To address these threats, we revisit their assumptions. We identify the purpose of data processing and explore the potential application of PETs to enhance privacy. Where feasible, appropriate PETs are incorporated.

**A. Preparation by applying privacy strategies.** Before analyzing the assumptions and phases relevant to PET selection, we adopted the following general strategies (where applicable): i) *Minimize*: We revisited the original assumptions, asking whether the data in question was truly necessary (cf. Sect. 4). For age verification, the application of Attribute-Based Credentials (ABCs) could be considered. ii) *Hide*: Encrypt all collected data at rest (e. g., disk/database encryption) and in transit (e. g., TLS); ii) *Enforce*: Implement strict access control (e. g., role-based) to safeguard data and ensure auditability; iv) *Inform*: Provide users with clear and accessible information about data processing and its purposes, such as through a privacy policy, data collection notices, and regular updates; v)

*Control*: Enable users to manage their preferences, and access, delete, or update their personal information — via a user dashboard, data deletion protocols, opt-in mechanisms, and consent withdrawal.

**B. PET selection process.** To identify additional potential PETs, we examined the data items used in each phase. Table 1 provides an overview of how data is used across phases. For example, one result of this activity was the identification of homomorphic encryption as a potential PET for encrypting location, time, and route data of vehicles, thereby enabling vehicle allocation while preserving confidentiality and still allowing matching with the (also encrypted) user location.

**C. Threat assessment.** We conducted an additional LINDDUN analysis using the revised assumptions. The revised assumptions have the potential to mitigate or eliminate most of the previously identified threats. However, we were unable to eliminate threats regarding linkability and identifiability (LINDDUN threats L.1.1, I.1.1, and I.2.2.1), as these stem from the use of a unique identifier. Nevertheless, for the purposes of our use case, it does not constitute a privacy problem if the *SP* can identify a *User*. It is important to note that even if advanced PETs (e.g., attribute-based credentials, zero knowledge proofs, anonymous payment) are implemented to allow anonymous use of the service, the *SP* may still be able to identify a user through data correlation (e.g., pick-up/drop-off locations, routes, and times), behavioral patterns, or service customization. Furthermore, in certain jurisdictions, the *SP* may be obligated to collect specific information for legal compliance, making full anonymity impossible.

**Further Limitations.** The main limitation of this approach is that it is not a systematic methodology. We first identified suitable privacy strategies following Hoepman [13], and then mapped them to relevant PETs. However, Hoepman’s strategies are defined at a higher level than PET Selection. As a result, we analyzed assumptions and determined the deployability of specific PETs to address certain threats based on our own experience, without a formal method. This introduces two limitations: i) The approach requires experienced experts to produce useful results, and ii) Different teams may reach different conclusions, reducing consistency and repeatability.

## 6 Analysis of PET Selection Approaches

In this section, we analyze the findings from the three PET selection attempts of Sections 5.1-5.3, highlighting their respective strengths and weaknesses. Table 2 provides a comparative summary of our analysis. We also extract insights to guide future research on PET selection methodologies.

### 6.1 Strengths

Each of the methodologies considered (Sect. 5.1-5.3) has its own strengths, which are largely complementary.

Table 2: Comparison of PET Selection Approaches

Criterion	Kunz et al. (2020)	Kunz & Binder (2022)	Al-Momani et al. (2022)
Core Method	Data- and service-driven filtering of PETs	PET matrix by goal, scenario, maturity, impact	Decision trees linking LINDDUN threats to strategies
Design Stage Fit	Assumes mature design, known data	Requires detailed threats	Needs mapped assumptions and threats
Final PET Selection Support	No decision logic for choosing among PETs	Maturity/impact noted but no guidance	No prioritization among PETs
Scalability / Use Case Fit	Too granular for large systems	Partial threat coverage	Partial PET coverage; requires expert tuning
Handles Context	Recognizes variation but lacks structured support	Treats PET effects as static across contexts	Accounts for necessity of data
Threat Interdependency	Treats threats independently	Treats threats independently	Considers shared assumptions, but not systematically
PET Coverage	Narrow focus on obfuscation PETs	Moderate PET list with missing types	Incomplete list (e.g., omits TLS, ZKPs, ABCs)
Strengths	Combines data/service analysis; domain-specific taxonomy	Maturity and impact dimensions included	Leverages threat rationale; supports strategy prioritization
Limitations	High effort; limited guidance for final PET selection	Ambiguous threat-to-PET mapping; lacks detail on PET variants	Limited PET set; lacks automation or consistency

The approach of Kunz et al. [20] promotes a combination of data-driven and service-driven elicitation. This is a sensible idea, as both the characteristics of the data and the requirements of the service influence the set of applicable PETs. The paper also introduces the concept of a domain-specific data taxonomy, with a set of applicable PETs mapped to each identified data type. This is an interesting idea that could help make PET selection more efficient.

The approach of Kunz and Binder [21] considers PET maturity as well as the impact of PETs on performance, architecture, and utility. Each of these aspects may be important in practice.

The approach of Al-Momani et al. [3] leverages detailed threat assessment information when selecting PETs. Our experience confirmed the value of this idea: the threat assessment improved our understanding of the origins and potential consequences of privacy threats, which proved helpful for PET selection.

## 6.2 Weaknesses

As described in Sect. 5, applying each of these academic approaches to our use case was problematic. Beyond the specific weaknesses of individual approaches, which may reflect their relative immaturity, we encountered several recurring limitations that may indicate more fundamental limitations. First, each approach seems to assume a completed system design. However, by that point, introducing PETs may be too late, as they could potentially impact core design choices. None



of the approaches supports an agile process in which the general system design and privacy considerations evolve in parallel, influencing each other iteratively.

Second, each approach assumes a fixed list of PETs and clear criteria for applicability. In practice, PET lists are often arbitrary, and the applicability of a given PET typically depends on context. Determining the impact of a PET (e.g., on performance, architecture, functionality, or future extensibility) requires careful analysis and substantial design effort [23]. The reviewed approaches tend to overlook this and rely on over-simplified generalizations.

Third, while existing approaches may identify potentially applicable PETs, they offer little guidance for making a final selection. This gap is especially critical in scenarios with specific accuracy and performance requirements. For example, when adding noise, it should sufficiently obscure privacy-relevant information without degrading the utility of the data. The performance impact of a PET also depends on the context: real-time applications impose stricter constraints than offline or batch-processing tasks. Moreover, the outcome depends not only on the PET itself but also on its configuration (e.g., the  $\epsilon$  value in differential privacy).

Fourth, each approach treats threats in isolation, selecting at least one PET per threat. In reality, both threats and PETs may be interdependent. For example, a single PET might mitigate multiple threats, or the use of one PET could interfere with the effectiveness of another. Focusing solely on local decisions can lead to overall suboptimal or even infeasible outcomes.

Finally, each approach omits considerations that fall outside their defined scope, such as “soft privacy” goals or security requirements. While this is understandable in a research setting, practical methodologies must be more comprehensive to be useful in real-world deployments.

### 6.3 Recommendations for Future Methodology

Insights from the pragmatic approach could help inform the development of improved methodologies. We offer the following recommendations.

**Investigate Assumptions.** When identifying mitigation techniques, we found it important to trace threats back to their underlying causes. The origin of a threat often constrains the available mitigation options. For example, if Identifiability threats arise due to legal requirements to identify users, then PETs that provide anonymity may not be applicable. To support this process, we found it useful to document data protection-related assumptions about the system and to link each identified threat to the assumptions that give rise to it. This also helped identify cases where multiple threats stemmed from a shared assumption, meaning that a single PET targeting that assumption could address several threats. Revisiting assumptions and clarifying the purpose of data processing proved to be a valuable step in preparing for PET selection.

**Specific Step-wise Dataflows.** Structuring the use case into discrete steps helped streamline PET selection. It allowed us to visualize when and where data is created, to identify dependencies, and to avoid unintended side effects when applying PETs. A PET applied to mitigate a threat in one step may influence other steps where the same data is used.

**PETs' Appropriateness.** Addressing the limitations of current approaches will require improved support for selecting PETs in specific scenarios. In particular, new methodologies should help map scenario-specific requirements to the expected changes in system properties (e.g., performance, accuracy) resulting from the implementation and configuration of PETs. This would inevitably bring deployment and integration changes to the system that should be investigated by new methodologies.

**Adaption to Design Phase.** Different phases of the system design process require distinct tools and approaches. Designing a system from scratch allows building privacy into the architecture from the ground up. In contrast, improving an existing system demands a detailed understanding of current data flows to assess whether introducing a PET is feasible. For example, adding noise to encrypted data is not straightforward and may compromise functionality. Introducing a PET might also disrupt operations if essential data becomes inaccessible. If the system incorporates machine learning, additional considerations arise, such as the distinction between the initial training phase and the deployment of the model, which may affect how and when PETs can be applied.

**Addressing Compliance.** None of the approaches considered compliance. A future approach for PET selection could aim to bridge the gap between building privacy-friendly systems and ensuring regulatory compliance. Aligning privacy engineering with compliance requirements would significantly improve practical adoption. This is especially relevant in corporate environments, where privacy processes are often structured around meeting legal and regulatory standards.

## 7 Conclusions and Future Work

The PET selection methods found in the literature exhibit significant shortcomings. While they offer some guidance, they often rely on oversimplified assumptions (e.g., regarding the applicability of a PET in a given situation), and fall short of providing a complete methodology. In some cases, these approaches yield a list of potentially applicable PETs, but the challenge of selecting the most appropriate one remains. This requires evaluating the maturity of each PET, its compatibility with performance and architectural constraints, the availability of ready-to-use implementations etc.

The pragmatic approach presented in this paper cannot be considered a methodology in its current form, as it heavily relies on the expertise of the team. The challenge of selecting appropriate PETs remains open, and current approaches can only partially support this task.

Our work highlights the importance of using realistic use cases for evaluating PET selection methodologies. Post-ride actions, such as service enhancements or monetization, can directly influence PET selection. For example, issuing invoices must comply with legal requirements regarding the included data.

While our analysis highlights the challenges of selecting PETs in real-world scenarios, it does not offer a complete solution. Even after PETs are selected, implementing, integrating, and configuring them remains a significant challenge

[12]. There is a need for more iterative, agile, and exploratory approaches that support "what-if" analysis, allowing design teams to evaluate the impact of selected PETs without immediate commitment. Privacy should be integrated into overall system design, not treated as a separate, downstream process. The use of Artificial Intelligence techniques to support PET selection also represents a potential direction for future work.

**Acknowledgments.** This work was inspired by privacy engineering discussions at Dagstuhl Seminar 23242, "Privacy Protection of Automated and Self-Driving Vehicles". The work was supported in part by the U.S. National Science Foundation (NSF) under grant number 2245323, and by the German Federal Ministry of Education and Research (BMBF) under grant number 16KIS1382.

## Bibliography

- [1] Adams, C.: Introduction to Privacy Enhancing Technologies: A Classification-Based Approach to Understanding PETs. Springer (2021)
- [2] Al-Momani, A., Balenson, D., Mann, Z.Á., Pape, S., Petit, J., Bösch, C.: Navigating privacy patterns in the era of robotaxis. In: IEEE European Symposium on Security and Privacy Workshops, pp. 32–39, IEEE (2024)
- [3] Al-Momani, A., Bösch, C., Wuyts, K., Sion, L., Joosen, W., Kargl, F.: Mitigation lost in translation: leveraging threat information to improve privacy solution selection. In: ACM SAC (2022)
- [4] Alhirabi, N., Beaumont, S., Rana, O., Perera, C.: Designing privacy-aware iot for unregulated domains. ACM Transactions on Internet of Things (2023)
- [5] Baldassarre, M.T., Barletta, V.S., Caivano, D., Scalera, M.: Integrating security and privacy in software development. Software Quality Journal **28**(3), 987–1018 (2020)
- [6] Bellotti, V., Sellen, A.: Design for privacy in ubiquitous computing environments. In: ECSCW, pp. 77–92, Springer (1993)
- [7] Chah, B., Lombard, A., Bkakra, A., Yaich, R., Abbas-Turki, A., Galland, S.: Privacy threat analysis for connected and autonomous vehicles. Procedia Computer Science **210**, 36–44 (2022)
- [8] Deng, M., Wuyts, K., Scandariato, R., Preneel, B., Joosen, W.: A privacy threat analysis framework: supporting the elicitation and fulfillment of privacy requirements. Requirements Engineering **16**(1), 3–32 (2011)
- [9] Dev, J., Rashidi, B., Garg, V.: Models of applied privacy (map): A persona based approach to threat modeling. In: ACM CHI, pp. 1–15 (2023)
- [10] Drozd, O.: Privacy pattern catalogue: A tool for integrating privacy principles of iso/iec 29100 into the software development process. Privacy and Identity Management pp. 129–140 (2016)
- [11] Gürses, S., Troncoso, C., Diaz, C.: Engineering privacy by design. Computers, Privacy & Data Protection **14**(3), 25 (2011)

- [12] Herwanto, G.B., Ekaputra, F.J., Quirchmayr, G., Tjoa, A.M.: Towards a holistic privacy requirements engineering process: Insights from a systematic literature review. *IEEE Access* (2024)
- [13] Hoepman, J.: Privacy design strategies - (extended abstract). In: *ICT Systems Security and Privacy Protection SEC, IFIP AICT*, vol. 428 (2014)
- [14] Hong, J.I., Ng, J.D., Lederer, S., Landay, J.A.: Privacy risk models for designing privacy-sensitive ubiquitous computing systems. In: *ACM DIS*, pp. 91–100 (2004)
- [15] Jensen, C., Tullio, J., Potts, C., Mynatt, E.D.: Strap: a structured analysis framework for privacy. *Georgia Institute of Technology* **1** (2005)
- [16] Jordan, S., Fontaine, C., Hendricks-Sturup, R.: Selecting privacy-enhancing technologies for managing health data use. *Frontiers in Public Health* **10**, 814163 (2022)
- [17] Kalloniatis, C., Kavakli, E., Gritzalis, S.: Addressing privacy requirements in system design: the pris method. *Requirements Engineering* **13** (2008)
- [18] Katcher, S., Ballard, B., Bloom, C., Isaacson, K., McEwen, J., Shapiro, S., Slotter, S., Paes, M., Xu, R.: The mitre panoptic™ privacy threat model tutorial. In: *2nd Workshop on Privacy Threat Modeling (WPTM)* (2023)
- [19] Kung, A.: Pears: Privacy enhancing architectures. In: *Privacy Technologies and Policy - 2nd Annual Privacy Forum (APF)*, pp. 18–29 (2014)
- [20] Kunz, I., Banse, C., Stephanow, P.: Selecting privacy enhancing technologies for iot-based services. In: *EAI SecureComm*, pp. 455–474 (2020)
- [21] Kunz, I., Binder, A.: Application-oriented selection of privacy enhancing technologies. In: *Privacy Technologies and Policy - 10th Annual Privacy Forum, APF, LNCS*, vol. 13279, pp. 75–87, Springer (2022)
- [22] Löbner, S., Tronnier, F., Pape, S., Rannenber, K.: Comparison of de-identification techniques for privacy preserving data analysis in vehicular data sharing. In: *ACM CSCS*, pp. 7:1–7:11, ACM (11 2021)
- [23] Mann, Z.Á., Petit, J., Thornton, S.M., Buchholz, M., Millar, J.: SPIDER: Interplay assessment method for privacy and other values. In: *2024 IEEE European Symposium on Security and Privacy Workshops (EuroS&PW)*, pp. 1–8, IEEE (2024)
- [24] Pape, S., Bkakra, A., Chah, B., Heymann, M., Winkler, S.S.: A framework for supporting PET selection based on GDPR principles. In: *ARES* (2025)
- [25] Pape, S., Rannenber, K.: Applying privacy patterns to the internet of things’ (IoT) architecture. *Mobile Networks and Applications* (2019)
- [26] Pape, S., Syed-Winkler, S., Garcia, A.M., Chah, B., Bkakra, A., Hiller, M., Walcher, T., Lombard, A., Abbas-Turki, A., Yaich, R.: A systematic approach for automotive privacy management. In: *ACM CSCS* (2023)
- [27] de Vries, R., Mann, Z.Á.: Secure neural network inference as a service with resource-constrained clients. In: *Proceedings of the IEEE/ACM 16th International Conference on Utility and Cloud Computing* (2023)
- [28] Wuyts, K., Joosen, W.: Linddun privacy threat modeling: a tutorial. *CW Reports* (2015)

# Performance Analysis of Lightweight Transformer Models for Healthcare Application Privacy Threat Detection

Jude E. Ameh<sup>1</sup>[0009-0001-4523-5204], Abayomi Otebolaku<sup>1</sup>[0000-0002-4320-9061], Alex Shenfield<sup>1</sup>[0000-0002-2931-8077], Augustine Ikpehai<sup>1</sup>[0000-0002-5254-8188] & Dauda Sule<sup>2</sup>[0000-0002-8795-4717]

<sup>1</sup> Sheffield Hallam University  
Sheffield, United Kingdom  
<sup>2</sup> Air Force Institute of Technology  
Kaduna, Nigeria  
j.e.ameh@shu.ac.uk

**Abstract.** The growing complexity of cyber threats in healthcare demands advanced, computationally efficient security solutions. This study employs a white-box approach to evaluate lightweight transformer models for detecting privacy threats in C/C++ healthcare software. We introduce a novel dataset annotated with privacy vulnerabilities using the LINDDUN methodology, covering linkability, identifiability, non-repudiation, detectability, information disclosure, unawareness, and non-compliance. A systematic mapping between LINDDUN threats and Common Weakness Enumeration (CWE) classifications standardize privacy risk assessment. Six lightweight transformer models—GraphCodeBERT-base, CodeGPT-small, BERT-base-uncased, DistilRoBERTa-base, DistilBERT-base, and T5-small were fine-tuned and evaluated on the dataset containing 56,395 vulnerable and 364,232 non-vulnerable C/C++ functions, sourced from open-source projects to mitigate coder bias. All models achieve over 98% accuracy, with T5-small reaching 98.64%. Detailed computational costs, including model parameters and training times (~12 hours), highlight suitability for resource-constrained environments. This work validates NLP-driven privacy risk assessment, offering a scalable framework for healthcare security.

**Keywords:** Healthcare privacy, lightweight, LINDDUN framework, Software vulnerability detection, Privacy threat modelling

## 1 Introduction

Healthcare organizations face increasing cyberattacks, such as the 2017 WannaCry ransomware outbreak that caused unprecedented disruptions (Portela et al., 2023). Traditional security approaches like signature-based detection find it difficult to detect advanced persistent threats (Dequino et al., 2025), thus necessitating efficient and novel detection strategies.

Natural language processing (NLP) advancements which are enabled by transformer-based models, offer new vulnerability detection possibilities, but state-of-the-

art transformer models with billions of parameters create high computational costs and substantial memory requirements (Latharani & Mouneshachari, 2024; Denecke et al., 2024). While effective at analyzing unstructured data for security risks, the computational demands of these models hinder deployment in resource-constrained environments like medical devices (Thapa et al., 2022).

This study evaluates lightweight transformer models for detecting privacy threats in healthcare software, focusing on real-time, computationally efficient solutions. We introduce a novel C/C++ code dataset annotated with privacy vulnerabilities using the LINDDUN privacy threat methodology, which categorizes threats into Linking, Identifying, Non-repudiation, Detecting, Data Disclosure, Unawareness, and Non-compliance (Wuyts & Joosen, 2020). We establish a systematic mapping between LINDDUN categories and Common Weakness Enumeration (CWE) classifications (Lohmann, Albuquerque, & Machado, 2023). C/C++ was selected due to it is considered a programming language for safety-critical systems (Zouev, 2020), and its manual memory management introduces unique privacy vulnerabilities like buffer overflows (Pereira et al., 2021) which align with LINDDUN categories and can cause unauthorized data exposure (Li et al., 2023). Hence, this focus addresses a research gap, as existing datasets often prioritize general security over privacy-specific vulnerabilities in healthcare (Wuyts & Joosen, 2020).

This research contributes: (1) a novel healthcare-specific C/C++ dataset annotated with LINDDUN-based privacy vulnerabilities, (2) a systematic LINDDUN-CWE mapping framework that integrates privacy risk assessment with software security analysis, and (3) a comprehensive evaluation of lightweight transformer models for privacy threat detection. These advancements promote privacy-aware security while ensuring computational efficiency, useful for scalable, AI-driven security solutions in healthcare.

The remaining sections of this paper are organized as follows. Section II provides a background and gives further insights by showcasing related works. Section III provides a concise methodology of the methods, approach and experiments performed to achieve the objectives of this paper. While Section IV, V, and VI showcase the results of the experiments, provide a critical analysis in a discussion and conclusion respectively.

## 2 Background and Related Works

The digitization of healthcare has revolutionized medical services, enhancing patient outcomes and administrative efficiency. However, this transformation has introduced significant challenges in data privacy, security vulnerabilities, and interoperability, which now require advanced analytical frameworks and computationally efficient threat detection models (Ahmed et al., 2023; Silva et al., 2024).

### 2.1 Healthcare Information Systems and Data Privacy

Modern healthcare information systems are built upon intricate networks of stakeholders and information systems, where Electronic Health Records (EHRs) have evidently enhanced clinical decision-making and patient outcomes (Alomar et al., 2024).

However, persistent system fragmentation and dependence on proprietary data formats continue to impede interoperability, complicating secure and efficient data exchange among disparate platforms (Holmgren, Everson & Adler-Milstein, 2022). Standardized frameworks such as Health Level Seven Fast Healthcare Interoperability Resources (HL7 FHIR) and ISO/EN 13606 offer blueprints for harmonized data structures, but variable implementation practices undermine their potential for seamless integration across institutions (Salunkhe et al., 2024).

The migration of healthcare workloads to cloud environments delivers significant gains in scalability and resource optimization but simultaneously introduces elevated privacy and compliance risks, including unauthorized data access and multitenancy concerns (Sivan, R. and Zukarnain, 2021). As providers increasingly harness artificial intelligence and big-data analytics to inform diagnostics and operational workflows, questions around data ownership, informed consent procedures, and algorithmic transparency have become critical ethical and legal considerations (Karimian et al., 2022; Solanki et al., 2022). Moreover, the healthcare sector faces a growing spectrum of cybersecurity threats such as ransomware and distributed denial-of-service attacks, and insider exploits, that increase existing vulnerabilities. Traditional cryptographic safeguards often prove insufficient against sophisticated, persistent adversaries, while machine learning-powered decision-support systems remain susceptible to adversarial manipulation, underscoring the urgent need for advanced privacy protections and resilient threat-detection models (Cinà et al., 2023).

## 2.2 Privacy Threat Modelling with LINDDUN

To systematically address privacy risks during system design, researchers have developed specialized threat modelling frameworks. LINDDUN is a prominent privacy threat modelling methodology that provides a structured approach to identify and mitigate privacy threats in software architectures. Deng et al. (2011) introduced LINDDUN as the privacy counterpart to STRIDE of Microsoft security model. By analyzing data flow diagrams of a system, LINDDUN guides analysts to consider how each component or data flow could be subject to the seven types of privacy threats. For example, linkability checks if an attacker could link two pieces of data (or events) to the same person, while non-compliance examines whether the system might violate privacy laws or policies.

LINDDUN has gained wide recognition as a robust framework for privacy-by-design. Acknowledged by the NIST Privacy Framework<sup>1</sup>. It is a strong methodology for evaluating privacy risks, it is particularly relevant in healthcare, where continuous exchange of sensitive patient data demands rigorous threat assessment. For example, LINDDUN enables the identification of threats like linkability and identifiability in EHR systems, ensuring compliance with regulations such as the General Data Protection Regulation (GDPR) (Wuyts & Joosen, 2020).

Overall, LINDDUN serves as a foundation for our methodology, providing a systematic method to examine how privacy can be violated in healthcare software. By

<sup>1</sup> <https://www.nist.gov/privacy-framework/linddun-privacy-threat-modeling-framework>

acknowledging its limitations and augmenting it with risk-based filtering and CWE mappings, harnessing the broad coverage while maintaining practical relevance.

Despite its strengths, LINDDUN has limits, such as the "threat explosion" problem, where extreme threat identification engulfs resources (Robles-González et al., 2020).

### 2.3 Lightweight Transformer Models

A review of recent research has revealed a growing interest in transformer-based deep learning models in software security tasks including vulnerability detection, code review automation, and malware analysis (Thapa et al., 2022). Transformer-based language models, originally developed for NLP tasks, have proven exceptionally adept at understanding source code because code has structural similarities to natural language (it follows grammatical rules and has context-dependent semantics). When fine-tuned, these models can detect subtle bugs or vulnerabilities that might elude manual code inspection. For example, Thapa et al. (2022) demonstrated that transformers fine-tuned on a corpus of vulnerable code can achieve high recall in detecting buffer overflows, pointer misuse, and other C/C++ vulnerabilities, significantly outperforming traditional machine learning classifiers.

However, the limitation of these powerful models is their computational complexity. A standard transformer like BERT-base has 110 million parameters and requires considerable memory and processing time for inference. Lightweight transformers using techniques such as knowledge distillation, parameter pruning, and quantization are used to compress models while trying to retain most of their accuracy (Dantas et al., 2024). Sanh et al. (2019) pioneered this with DistilBERT, showing that a model with almost half the parameters of BERT could retain ~97% of the language understanding capabilities by learning from outputs of BERT during training. Similarly, DistilRoBERTa was produced by distilling the RoBERTa model (a variant of BERT) and achieves comparable performance on many tasks with a fraction of the parameters.

Table 1 summarizes some characteristics of lightweight transformer models relevant to this work, including their size reductions and design strategies. Full versions of these models have been successfully applied to security tasks in prior code specific research (Fernando et al., 2020; Guo et al., 2021). Even with these models achieving state-of-the-art results on code understanding benchmarks and vulnerability classification tasks, these models can be heavy and thus require smaller variants or further compression. Luo et al. (2023) presents a study on optimizing transformer models for resource-constrained environments, highlighting that methods like layer pruning (removing some transformer layers) and weight quantization (reducing precision) can significantly speed up inference with minimal loss of accuracy.

Finally, while transformers can flag patterns correlating with vulnerabilities, they tend to be "black boxes." For adoption in regulated industries like healthcare, the explainability of model decisions is important (Alkhanbouli et al., 2025). There is growing interest in explainable AI for security, e.g. highlighting code lines that influenced the prediction of the model (Marey et al., 2024). This is somewhat outside the scope of our current work, but we acknowledge it as an important direction for making ML-driven security tools more transparent to auditors and developers.



**Table 1.** Lightweight transformer models used, including compression methods and parameter counts (to be presented in results).

Model	Original Size (Parameters)	Compressed Size (Parameters)	% Reduction	Method of Compression	Efficiency Improvements
BERT-base-uncased	125M	67M	46%	Knowledge Distillation	Maintains strong performance on code understanding tasks
GraphCodeBERT-Base	125M	66M	47%	Parameter Sharing & Layer Pruning	Optimized for faster inference and lower memory usage
CodeGPT-Small	124M	65M	48%	Reduced Transformer Layers	Enables efficient code generation and completion
T5-Small	220M	110M	50%	Knowledge Distillation & Pruning	Similar performance to full-sized counterpart with improved efficiency
DistilRoBERTa-Base	355M	134M	62%	Knowledge Distillation	40% reduction in parameters and faster inference
DistilBERT-Base	110M	66M	40%	Knowledge Distillation	Nearly same performance as BERT with 40% parameter reduction

## 2.4 Healthcare Security Datasets

Effective privacy threat detection relies on high-quality, domain-specific datasets. In the domain of software vulnerability detection, several datasets have been proposed in recent years, but few focus on the healthcare context or on privacy threats specifically. However, the AI4HEALTHSEC dataset is one that aggregates threat intelligence from medical software vulnerabilities and hospital security incidents, providing a foundation for healthcare cybersecurity research (Silvestri et al., 2023). However, the focus of such threat intelligence datasets is often on unstructured data (textual reports, logs) rather than code. DiverseVul and ReposVul datasets, are general code centric sources that offer comprehensive collections of C/C++ vulnerabilities, with 18,945 vulnerable functions and repository-level tracking, respectively (Li et al., 2023; Wang et al., 2024).

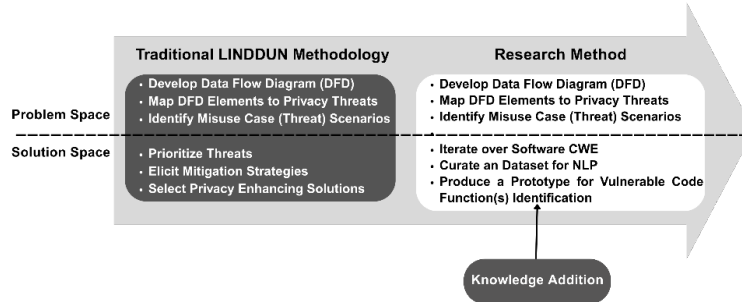
Furthermore, challenges such as data imbalance and limited generalization persist, prompting research into automated dataset augmentation techniques (Thabtah et al., 2020). Privacy-specific datasets for healthcare are particularly scarce, as most existing datasets focus on general security concerns (Silva et al., 2024). This gap highlights the need for specialized datasets tailored to healthcare privacy threats.

## 2.5 Research Gap and Contributions

The integration of LINDDUN privacy threat modeling, lightweight transformer models, and healthcare-specific security datasets presents a promising yet underexplored direction for privacy threat detection (Wuyts & Joosen, 2020; Thapa et al.,

2022). While LINDDUN provides systematic threat assessment, lightweight transformers enable efficient analysis, and specialized datasets offer domain-specific training data, their combined potential remains largely untapped in healthcare cybersecurity (Silva et al., 2024). This study addresses this gap by synthesizing these components into a cohesive framework, as illustrated in Figure 1, which outlines the novelty and contributions of our approach. By leaning on a LINDDUN-annotated C/C++ dataset, a LINDDUN-CWE mapping, and lightweight transformer models, this research advances practical, efficient, and comprehensive privacy threat detection mechanisms for healthcare environments facing increasingly sophisticated cyber threats.

To the best of our knowledge, prior to this work there was no publicly available dataset that labels code explicitly with privacy threat categories (LINDDUN or similar). Our approach can be seen as synthesizing a privacy-focused dataset by filtering existing vulnerabilities through the lens of a privacy threat model. The result is a dataset where each vulnerable example is not just a random bug, but one that maps to a privacy threat. We provide details of this mapping in the methodology section.



**Fig. 1.** Conceptual framework illustrates the extension of LINDDUN privacy threat modelling with CWE.

### 3 Methodology

Our research methodology is made up of four key components: (i) privacy threat modelling using LINDDUN to identify potential privacy threats in a healthcare system, (ii) mapping those threats to software weakness types (CWEs) and constructing a labelled code dataset, (iii) selection and implementation of lightweight transformer models for vulnerability detection, and (iv) evaluating the performance and efficiency of the selected models. Figure 1 illustrates the workflow, starting from system modelling and threat analysis, through data annotation, to model training and evaluation.

#### 3.1 System Modelling and Privacy Threat Analysis using LINDDUN

The modelling began with constructing high-level Data Flow Diagram (DFD) to represent patient journeys through healthcare facilities, from registration to follow-up care (Wuyts & Joosen, 2020).

The proposed method adapts the traditional LINDDUN methodology into three core steps, as illustrated in Figure 1:

(a) System Modelling: Creating a high-level DFD to map entities (E, e.g., patients, medical staff), processes (P, e.g., EHR systems), data flows (DF, e.g., user streams), and data stores (DS, e.g., databases). (b) Privacy Threat Identification: Iteratively analyzing DFD elements, using threat trees, for privacy threats (use- and mis- cases) using the seven LINDDUN categories and, (c) Threat Mapping: Linking identified threats to Common Weakness Enumeration (CWE) categories to standardize vulnerabilities.

This unique methodology extends LINDDUN by mapping identified threats to CWE categories (e.g., linkability to CWE-200: Information Exposure), creating a novel bridge between privacy and software vulnerabilities. The high level DFD of a Healthcare Information System (HIS) capturing patient interactions, Threat trees (documenting use and misuse cases), and final mappings to CWE categories are provided here<sup>2</sup>. This framework supports subsequent model development by ensuring precise identification of privacy threats.

### 3.2 Dataset Construction and Integration

The dataset was developed by integrating LINDDUN-based privacy threat annotations with two established vulnerability datasets: DiverseVul and ReposVul (Li et al., 2023; Wang et al., 2024). DiverseVul contains 18,945 vulnerable C/C++ functions across 150 Common Weakness Enumeration (CWE) types, sourced from multiple open-source projects contributed by diverse developers from fields such as software engineering, cybersecurity, and healthcare (Li et al., 2023). ReposVul, the first dataset to implement repository-level vulnerability tracking, includes code from varied open-source repositories across domains like web development, embedded systems, and medical software, authored by developers with diverse expertise (Wang et al., 2024). This diversity in contributors and project domains ensures a broad representation of coding styles, reducing the risk of coder bias.

To further mitigate coder bias, code from both datasets was preprocessed using tokenization to standardize variable names, function signatures, and coding structures, neutralizing stylistic differences while preserving semantic content (Li et al., 2023). For example, variable names like `patient_id` and `userID` were normalized to generic tokens, ensuring models focus on structural vulnerabilities rather than superficial naming conventions, further diversifying the dataset and minimizing bias from localized coding practices (Silva et al., 2024).

The preprocessing pipeline merged DiverseVul and ReposVul with LINDDUN-based annotations, which were generated by mapping privacy threats (e.g., linkability, identifiability) to C/C++ functions using the methodology outlined in Section 3.1 (Wuyts & Joosen, 2020). A filtering process retained only functions aligned with privacy-relevant CWE categories, such as CWE-200 (Information Exposure) and CWE-327 (Broken Cryptography), ensuring relevance to healthcare privacy threats.

<sup>2</sup> <https://github.com/juxam/C3-VULMAP>

To illustrate the LINDDUN-CWE mapping process, consider the following examples of vulnerable C/C++ functions from our dataset:

Example 1 - Linkability Threat (CWE-200: Information Exposure):

```
c
void process_patient_data(char* patient_id, char* diagnosis) {
    printf("Processing: %s - %s\n", patient_id, diagnosis); // Vulnerability: Direct logging of patient identifiers enables linkability
}
```

Example 2 - Identifiability Threat (CWE-327: Broken Cryptography):

```
c
char* encrypt_patient_record(char* record) {
    // Vulnerability: Weak encryption allows patient re-identification
    return simple_xor_encrypt(record, "weakkey");
}
```

These examples demonstrate how specific coding patterns were mapped to LINDDUN categories, thereby providing concrete instances of privacy vulnerabilities that our models are trained to detect. Each function in our dataset included similar annotations linking code structure to privacy threat categories and thereby enabling systematic model training on privacy-specific patterns.

The final corpus comprised 56,395 vulnerable and 364,232 non-vulnerable C/C++ functions, balanced through random under sampling to address class imbalance (Thapa et al., 2022). This comprehensive dataset construction process found here<sup>3</sup>, ensures that models trained on this data are robust, generalizable, and tailored to real-world healthcare privacy vulnerabilities.

The semi-automated annotation process introduced potential subjectivity that may affect the reproducibility of the result. While the LINDDUN-CWE mapping provides systematic guidelines, the interpretation of specific code patterns as privacy threats required expert judgment, particularly for edge cases where vulnerability classification was ambiguous. To mitigate this limitation, a multi-reviewer annotation process was implemented where three security experts independently classified a subset of 5,000 functions, achieving an inter-rater reliability score (Cohen's  $\kappa$ ) of 0.78, indicating substantial agreement. However, annotation consistency challenges remain, particularly for context-dependent vulnerabilities where the privacy impact depends on broader system architecture or deployment scenarios.

### 3.3 Model Implementation

Six lightweight transformer models (GraphCodeBERT-base, CodeGPT-small, BERT-base-uncased, DistilRoBERTa-base, DistilBERT-base, and T5-small) were fine-tuned on our training dataset for binary vulnerability classification.

<sup>3</sup> <https://github.com/juxam/C3-VULMAP>

We employed AdamW optimizer with 0.01 weight decay and a linear learning rate scheduler with 10% warm-up steps. Learning rates were set to  $2e-5$  for BERT-based models and  $1e-4$  for CodeGPT/CodeT5 based on validation performance. Batch sizes were adjusted by model complexity: 32 for DistilBERT/DistilRoBERTa and 16 for others, with gradient accumulation when needed.

Training ran for up to 10 epochs with early stopping if validation F1 didn't improve for 2 consecutive epochs. We implemented data sampling where non-vulnerable examples were freshly sampled each epoch from a pool of  $\sim 300k$  examples, effectively providing data augmentation. Our balanced validation set ensured meaningful F1 scores during early stopping. All models were trained on RTX 3090 GPUs with mixed precision (FP16) to optimize memory usage. Training times varied by model complexity: DistilBERT/DistilRoBERTa ( $\sim 2$  hours), BERT/GraphCodeBERT ( $\sim 3$  hours), CodeGPT ( $\sim 4$  hours), and T5 ( $\sim 4.5$  hours). Each model used its specific tokenizer, with T5 reframing classification as text generation with classification prompts (Feng et al., 2020). Accuracy and F1 were tracked per epoch.

### 3.4 Performance Evaluation

Model performance was assessed using accuracy, precision, recall, and F1-score, validated via 5-fold cross-validation to ensure robustness (Chakraborty et al., 2021). Confusion matrices, labelled with 0 (non-vulnerable) and 1 (vulnerable), were generated to analyze model behavior. Computational efficiency was evaluated through epoch times, peak GPU memory usage, and model parameter counts, ensuring suitability for resource-constrained environments (Devlin et al., 2019). The evaluation compared the six lightweight transformer models, which were selected for their efficiency in processing structured and unstructured security data (Chakravarty & Haque, 2023).

## 4 Experimental Results

### 4.1 Dataset Composition and Significance

The dataset, comprising 56,395 vulnerable and 364,232 non-vulnerable C/C++ functions across 626 Common Weakness Enumeration (CWE) categories, is specifically designed for analyzing privacy risks in healthcare software, such as medical device firmware and EHR systems. The test set split consisted of approximately 60,000 samples (with a 3:1 non-vulnerable to vulnerable ratio, reflecting a realistic scenario). We ensured the test set contained examples across all seven LINDDUN threat categories. This allowed our models to be evaluated on their ability to detect vulnerabilities related to Linkability, Identifiability, etc., not just on a narrow subset. The diversity of this test set is important for assessing generalization. In a healthcare privacy context, missing a vulnerability that leads to, say, Non-compliance (violating a legal requirement) could be just as serious as missing one that leads to Disclosure of information. Further, the test data included code never seen during training. Success on this test showed that the

model learned general patterns of vulnerabilities, rather than memorizing function specific cues.

## 4.2 Model Performance: Comparative Evaluation

The six lightweight transformer models were fine-tuned and achieved accuracy scores exceeding 98% via 5-fold cross-validation, affirming their suitability for healthcare privacy threat detection (Thapa et al., 2022). Table 2 presents performance metrics, with T5-small achieving the highest accuracy (98.64%), precision (98.54%), recall (98.64%), and F1-score (98.64%), alongside a validation loss of 0.0047, indicating superior generalization across privacy threat categories. GraphCodeBERT-base and CodeGPT-small, with training losses of 0.0412 and 0.0253, respectively, demonstrated faster convergence, likely due to their code-specific pre-training, which enhances structural and semantic pattern detection (Li et al., 2023).

Similarly, the best epochs, identified by peak F1-scores, ensured optimal comparisons. The low validation loss (0.0047) of T5-small suggests robust generalization, while the performance of GraphCodeBERT-base highlights its advantage in capturing code dependencies, critical for healthcare security audits (Li et al., 2023).

In figure 2 the confusion matrix for each model is shown and demonstrates the subtle distinctions in how precision and recall trade-offs manifest. For instance, GraphCodeBERT-base exhibited particularly strong sensitivity to subtle vulnerability patterns, correctly identifying 14,751 of 15,000 positive instances with only 245 false positives, whereas T5-small achieved the highest overall balance by correctly classifying 14,796 positives and yielding just 219 false positives. These matrices, normalized per actual class, show that models like DistilBERT-base and DistilRoBERTa-base maintain high true negative rates while slightly differing in false negative counts, reflecting their conservative detection strategies.

**Table 2.** Performance Metrics of Lightweight Transformer Models

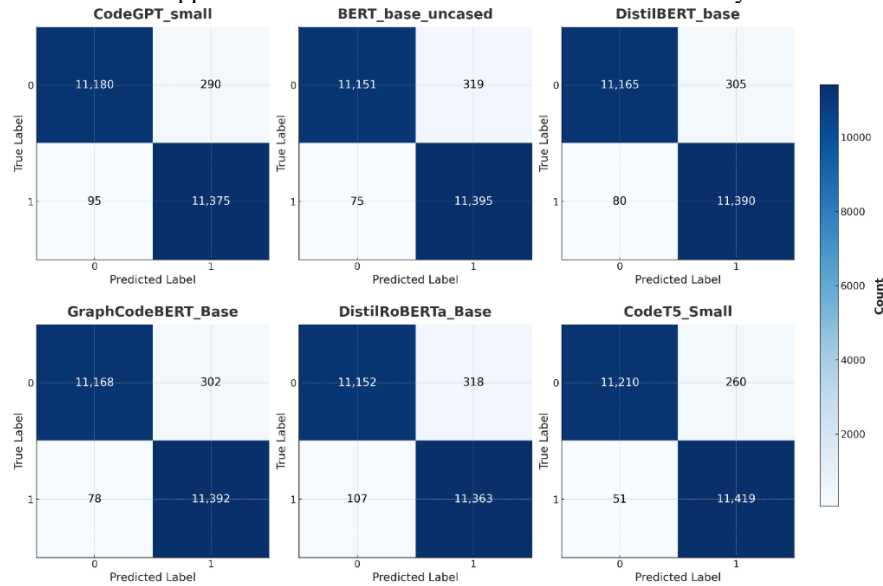
Model	Train Loss	Val Loss	Accuracy	Precision	Recall	F1 Score
GraphCodeBERT-base	0.0412	0.0610	0.9828	0.9836	0.9834	0.9834
CodeGPT-small	0.0253	0.0843	0.9832	0.9838	0.9836	0.9836
BERT-base-uncased	0.0336	0.0649	0.9589	0.9835	0.9832	0.9832
DistilRoBERTa-base	0.0439	0.0608	0.9765	0.9823	0.9819	0.9819
DistilBERT-base	0.0302	0.0744	0.9815	0.9834	0.9832	0.9832
T5-small	0.0029	0.0047	0.9864	0.9854	0.9864	0.9864

## 4.3 Comparison with Benchmarks

Our lightweight transformer models, achieving F1-scores above 98% on a healthcare-specific C/C++ dataset, appear to outperform recent benchmarks in vulnerability detection, particularly for privacy threats in healthcare applications. Li et al. (2023) evaluated large language models on a general-purpose dataset, with models achieving an F1-score

of approximately 58% and a recall of 87%. In contrast, our T5-small model recorded an F1-score of 98.64%, and DistilBERT-base reached 98.32%, suggesting superior performance in our targeted domain.

Zhou et al. (2019) applied BERT-based models to Python source code vulnerability detection, with DistilBERT achieving an F1-score of 0.92. Our base DistilBERT model, fine-tuned on healthcare C/C++ code, outperformed this with an F1-score of 0.9832. Chen and Monperrus (2021) reported F1-scores of 0.90–0.95 for a BERT based method on the SARD and Big-Vul datasets, which focus on general vulnerabilities. The higher Dataset and task differences, such as programming languages (C/C++ vs. Python) and focus (privacy vs. general vulnerabilities), limit direct comparisons. However, our results highlight the efficacy of our LINDDUN-CWE mapping and lightweight transformer architectures for healthcare cybersecurity, offering high accuracy and efficiency for real-world applications like medical device firmware and EHR systems.



**Fig. 2.** Confusion matrix of lightweight models trained on novel dataset (0: Non-vulnerable, 1: Vulnerable)

#### 4.4 Computational Efficiency and Resource Utilization

Table 3 details computational costs, including epoch times, peak GPU memory usage, and model parameters, critical for resource-constrained healthcare environments (Wang et al., 2019). DistilBERT-base was the fastest (0.13s/epoch) with 66M parameters, while DistilRoBERTa-base had the lowest memory footprint (1284.60 MB). CodeT5-small (60M parameters) balanced speed and memory, making it suitable for edge computing. CodeGPT-small, with higher resource demands (3777.08 MB), may be less practical for embedded systems.

**Table 3.** Computational Efficiency of Lightweight Transformer Models

Model	Epoch Time (s)	Peak GPU Memory (MB)	Parameters (M)
DistilRoBERTa-base	0.16	1284.60	134
DistilBERT-base	0.13	1536.28	66
T5-small	0.26	2467.19	60
GraphCodeBERT-base	0.33	2311.43	66
CodeGPT-small	0.36	3777.08	65
BERT-base-uncased	0.31	2822.02	110

These metrics, derived from mini-batch tests, highlight trade-offs between speed and memory, guiding model selection for specific healthcare deployment scenarios (Silva et al., 2024). For instance, the speed of DistilBERT-base suits real-time monitoring, while CodeT5-small shows a balance that supports adaptive threat detection.

However, these efficiency measurements were conducted under controlled laboratory conditions using high-end RTX 3090 GPUs with optimized software configurations, which may not accurately reflect real-world deployment challenges in healthcare environments. Healthcare institutions typically operate with heterogeneous hardware infrastructures, including legacy systems with limited computational resources or CPU-only environments where specialized accelerators are unavailable. Additionally, production deployments must contend with concurrent system loads, network latency constraints, and security overhead that can significantly impact inference times.

## 5 Discussion

The experimental results demonstrate the efficacy of lightweight transformer models in detecting privacy threats within healthcare software, with accuracies exceeding 98% on a novel C/C++ dataset annotated using the LINDDUN framework. Notably, T5-small achieved the highest accuracy of 98.64% and a validation loss of 0.0047, demonstrating its ability to generalize across critical privacy threat categories like linkability and identifiability, threats that empirical studies of healthcare apps have found to be prevalent (e.g., mental-health mobile apps often expose linkability and identifiability risks (Iwaya et al., 2023)). This finding extends prior work on transformer-based vulnerability detection by adapting general code analysis models to healthcare-specific privacy challenges (Ding et al., 2023). GraphCodeBERT-base (98.28%) and CodeGPT-small (98.32%), with training losses of 0.0412 and 0.0253 respectively, leverage their code-specific pre-training to excel in identifying structural and semantic features of privacy vulnerabilities. These rapid convergences of both models align with recent research showing that combining general large models with domain-adapted code models (e.g. CodeBERT/GraphCodeBERT) yields improved performance on specialized tasks (Sheng et al., 2024). Meanwhile, DistilBERT-base (98.15%) and DistilRoBERTa-base (97.65%) prioritize computational efficiency: as prior work notes, distilled models like DistilBERT are “smaller and faster” and retain much of the accuracy of larger BERT variants while halving runtime and model size (Wang et al., 2021). Such lightweight



models are thus ideal for resource-constrained healthcare environments (e.g. embedded medical devices) that require fast inference.

The LINDDUN-CWE mapping framework enhances the dataset by linking privacy threats to standardized software weaknesses, improving both model performance and interpretability – an essential requirement for healthcare regulatory compliance. This approach addresses a known gap in standardized threat taxonomies (Sheng et al., 2024). For instance, the strength of T5 small in capturing long-range dependencies aligns with the need to model adaptive threat scenarios, while the dataflow based pretraining of GraphCodeBERT-base provides structural insights useful for compliance audits (Sheng et al, 2024; Wang et al., 2021); together these suggest the potential of hybrid model architectures that combine sequence and graph encodings.

The remarkably high accuracy scores (>98%) achieved across all models raise important questions about dataset representativeness and real-world generalization. Our highly structured, LINDDUN-annotated dataset, while methodologically sound, may not adequately represent the complexity and variability of privacy vulnerabilities encountered in live healthcare environments. The systematic annotation process, though rigorous, creates a controlled experimental setting that may inflate performance metrics compared to deployment scenarios involving legacy code, mixed programming paradigms, or undocumented software components (Atiiq et al., 2024).

This concern is particularly relevant given the 3:1 ratio of non-vulnerable to vulnerable functions in our test set, which, while realistic, may not capture the long-tail distribution of rare but critical privacy vulnerabilities. The preprocessing steps that normalized coding styles and standardized variable names, while beneficial for reducing bias, may have inadvertently simplified the detection task by removing the stylistic complexity that models would encounter in real-world deployments. Future validation should include evaluation on unprocessed, production healthcare codebases to assess model robustness under more challenging conditions.

The reliance on static code analysis represents a fundamental limitation, as it cannot capture privacy vulnerabilities that emerge during runtime execution. Dynamic privacy threats in healthcare systems often manifest through network communications, user interaction patterns, or data processing workflows that are invisible to static analysis (Iwaya et al., 2023). For example, a function may appear secure in isolation but could leak patient information when combined with specific runtime configurations or when interacting with external APIs under certain conditions.

Healthcare systems frequently exhibit privacy violations through behavioural patterns including unauthorized data aggregation across multiple patient sessions, implicit inference attacks through query pattern analysis, or privacy breaches via timing side channels. These runtime phenomena require complementary dynamic analysis techniques such as runtime monitoring, network traffic analysis, and behavioural pattern detection. A comprehensive privacy threat detection framework should integrate both static code analysis with dynamic runtime monitoring to capture the full spectrum of privacy vulnerabilities in operational healthcare environments.

Another critical limitation of this work is the lack of model interpretability features, particularly concerning healthcare regulatory compliance. While our transformer models achieve high accuracy, they operate as "black boxes," providing limited insight into

decision-making processes. This opacity presents challenges for regulatory approval under frameworks like AI/ML guidance of the FDA, which emphasizes explainable AI for medical applications.

Future iterations should incorporate attention visualization techniques and gradient-based explanations to highlight code segments influencing vulnerability predictions. For instance, implementing SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations) could provide stakeholders with interpretable insights into model decisions. Such explainability features would enable security auditors to understand why specific code patterns trigger privacy threat classifications, supporting regulatory compliance and building trust among healthcare practitioners.

## 6 Conclusion and Future Work

This study contributes to the field of privacy-aware cybersecurity by introducing a novel healthcare software code dataset, annotated with privacy vulnerabilities based on the LINDDUN methodology, and by establishing a systematic mapping between LINDDUN threat categories and CWE classifications. The comprehensive evaluation of lightweight transformer models demonstrates that these models can achieve high accuracy, precision, recall, and F1 scores, often surpassing 98%, while maintaining computational efficiency suitable for real-world deployment. These findings validate the integration of NLP-driven analysis with structured privacy threat modelling, thereby providing a robust framework for automated privacy risk assessment in healthcare applications. The framework's efficiency supports deployment in resource-limited settings like electronic health record (EHR) systems and medical devices, addressing a pressing cybersecurity need in healthcare.

Looking ahead, several avenues for future research emerge. One promising direction is the expansion of the dataset to include multiple programming languages beyond C/C++, thereby broadening its applicability across diverse software ecosystems. Future work should also explore the integration of dynamic analysis techniques, such as runtime tracing and behavioral monitoring, to capture vulnerabilities that span interprocedural or repository-wide contexts. In addition, adversarial training methods could be incorporated to further enhance the robustness of transformer models against sophisticated, evasive privacy attacks. Finally, combining the strengths of multiple architectures, potentially through hybrid or ensemble approaches, could lead to even more effective privacy threat detection systems that balance generalizability with domain-specific feature extraction, paving the way for next-generation privacy-aware security solutions.

## 7 Limitations

The exclusive focus on C/C++ code represents a significant limitation that constrains the applicability of the framework across diverse healthcare software ecosystems. Modern healthcare applications are increasingly leveraged on web-based technologies (JavaScript, HTML5), high-level languages (Python for data analytics, Java for enterprise

systems), and mobile platforms (Swift, Kotlin) where privacy threats manifest differently. For instance, JavaScript applications may suffer from client-side data exposure through DOM manipulation or inadequate API sanitization, while Python applications might exhibit privacy violations through improper data serialization or inadequate access controls in machine learning pipelines.

Additionally, computational efficiency assessments were conducted under idealized laboratory conditions that may not represent actual deployment environments. Our measurements using RTX 3090 GPUs with optimized configurations provide upper-bound performance estimates, but healthcare institutions often operate with constrained, heterogeneous hardware configurations including legacy systems, shared computing resources, and security-hardened environments that introduce additional computational overhead. Real-world deployment factors such as concurrent system loads, thermal management, power constraints, and mandatory security processes could significantly impact the practical efficiency of these models, potentially requiring hardware upgrades or architectural modifications for acceptable performance in production healthcare settings.

**Acknowledgments.** The authors would like to express their gratitude to Sheffield Hallam University and the Air Force Institute of Technology for providing the research infrastructure and support necessary for this work. Special thanks to the cybersecurity research group at the College of Business, Technology and Engineering for their valuable feedback during the development of this research. We also acknowledge the support of the healthcare organizations that participated in our data collection efforts. This work was made possible through the collaborative efforts of researchers from both institutions.

**Disclosure of Interests.** The authors have no competing interests to declare that are relevant to the content of this article.

## References

1. Ahmed, M., Tushar, H., Thandi, N., & Seraj, R. (2023). Privacy-preserving AI in healthcare: Techniques and applications. *Comp. in Bio. & Me.*, 158, Article 106848.
2. Al Atiiq, S., Gehrmann, C., & Dahlén, K. (2024). Vulnerability detection in popular programming languages with language models. *arXiv:2412.15905*.
3. Alkhanbouli, R., Almadhaani, H. M. A., Alhosani, F., & Simsekler, M. C. E. (2025). The role of explainable artificial intelligence in disease prediction: A systematic literature review and future research directions. *BMC Medical Informatics and Decision Making*, 25, 110.
4. Chakraborty, S., Krishna, R., Ding, Y., & Ray, B. (2021). Deep learning based vulnerability detection: Are we there yet? *IEEE Transactions on Software Engineering*, 48(9), 3381-3397.
5. Chakravarty, S., & Haque, M. M. (2023). A comprehensive survey of deep learning in software engineering. *Software: Practice and Experience*, 53(10), 1897-1945.
6. Chen, Z., & Monperrus, M. (2021). A literature study of embeddings on source code. *Empirical Software Engineering*, 26(4), 1-35.
7. Cinà, A. E., Grosse, K., Demontis, A., Vascon, S., Zellinger, W., Moser, B. A., Oprea, A. M., Biggio, B., Pelillo, M., & Roli, F. (2023). Wild patterns reloaded: A survey of machine learning security against training data poisoning. *ACM Computing Surveys*, 55(13s).

8. Dantas, P. V., da Silva Jr., W. S., Cordeiro, L. C., & Carvalho, C. B. (2024). A comprehensive review of model compression techniques in machine learning. *Applied Intelligence*, 54, 11804-11844.
9. Deng, M., Wuyts, K., Scandariato, R., Preneel, B., & Joosen, W. (2010). A privacy threat analysis framework: Supporting the elicitation and fulfillment of privacy requirements. *Requirements Engineering*, 16(1), 3-32.
10. Denecke, K., May, R., & Rivera-R., O. (2024). Transformer models in healthcare: A survey & thematic analysis of potentials, shortcomings & risks. *Journal of Med. Systems*, 48(1).
11. Dequino, A., Bompani, L., Benini, L., & Conti, F. (2025). Optimizing BFloat16 deployment of tiny transformers on ultra-low-power extreme-edge SoCs. *Journal of Low Power Electronics and Applications*, 15(1), 8.
12. Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv:1810.04805*.
13. Ding, Y., Fu, Y., Ibrahim, O., Sitawarin, C., Chen, X., Alomair, B., Wagner, D., Ray, B., & Chen, Y. (2024). Vulnerability detection with code language models: How far are we? *arXiv:2403.18624*.
14. Feng, Z., Guo, D., Tang, D., Duan, N., Feng, X., Gong, M., Shou, L., Qin, B., Liu, T., Jiang, D., & Zhou, M. (2020). CodeBERT: A pre-trained model for programming and natural languages. In *Findings of the Assoc. for Compu. Linguistics: EMNLP 2020* (pp. 1536-1547). Association for Computational Linguistics.
15. Guo, D., Ren, S., Lu, S., Feng, Z., Tang, D., Liu, S., Zhou, L., Duan, N., Svyatkovskiy, A., Fu, S., Tufano, M., Deng, S. K., Clement, C., Drain, D., Sundaresan, N., Yin, J., Jiang, D., & Zhou, M. (2021). GraphCodeBERT: Pre-training code representations with data flow. In *International Conference on Learning Representations (ICLR 2021)*.
16. Holmgren, A. J., Everson, J., & Adler-Milstein, J. (2022). Assoc. of Hospital interoperable data sharing with alternative payment model participation. *JAMA Health Forum*, 3(2).
17. Iwaya, L. H., Babar, M. A., Rashid, A., & Wijayarathna, C. (2023). On the privacy of mental health apps: An empirical investigation and its implications for app development. *Empirical Software Engineering*, 28, Article 2.
18. Karimian, G., Petelos, E., & Evers, S. M. (2022). The ethical issues of the application of AI in Healthcare: A systematic scoping review. *AI and Ethics*, 2(4), 539-551.
19. Latharani, T. R., & Mouneshachari, S. (2024). Leveraging machine learning for behavioral analysis and mitigation of APT attacks in WSNs. *Journal of Electrical Systems*, 20(11S), 2174-2181.
20. Li, H., Ding, Z., Alowain, L., Chen, Y., & Wagner, D. (2023). DiverseVul: A new vulnerable source code dataset for deep learning based vulnerability detection. *arXiv:2304.00409*.
21. Li, H., Li, C., Wang, J., Yang, A., Ma, Z., Zhang, Z., & Hua, D. (2023). Review on security of Federated Learning and its application in Healthcare. *Future Generation Computer Systems*, 144, 271-290.
22. Lohmann, P. A., Albuquerque, C., & Machado, R. (2023). Systematic literature review of threat modeling concepts [Conference paper]. SCITEPRESS.
23. Luo, Z., Hanrui Yan, & Xueting Pan. (2023). Optimizing transformer models for resource-constrained environments: A study on model compression techniques. *Journal of Computational Methods in Engineering Applications*, 1-12.
24. Malihi, L., & Heidemann, G. (2023). Efficient & controllable model compression through sequential knowledge distillation and pruning. *Big Data & Cognitive Computing*, 7(3), 154.
25. Marey, A., Arjmand, P., Sabe Alerab, A. D., Eslami, M. J., Saad, A. M., Sanchez, N., ... & Umair, M. (2024). Explainability, transparency and black box challenges of AI in radiology:

- Impact on patient care in cardiovascular radiology. *Egyptian Journal of Radiology and Nuclear Medicine*, 55, Article 183.
26. Olomar, D., et al. (2024). The impact of patient access to Electronic Health Records on Health Care Engagement: Systematic Review. *Journal of Medical Internet Research*, 26.
  27. Portela, D., Nogueira-Leite, D., Almeida, R., & Cruz-Correia, R. (2023). Economic impact of a hospital cyberattack in a national health system: Descriptive case study. *JMIR Formative Research*, 7, e41738.
  28. Pereira, J. D., Ivaki, N., & Vieira, M. (2021). Characterizing buffer overflow vulnerabilities in large C/C++ projects. *IEEE Access*, 9, 142879–142892. DOI 10.1109/ACCESS.2021.3120349
  29. Robles-González, A., Parra-Arnau, J., & Forné, J. (2020). A LINDDUN-based framework for privacy threat analysis on identification and authentication processes. *Computers & Security*, 94, Article 101755.
  30. Salunkhe, V., et al. (2024). EHR interoperability challenges leveraging HL7 FHIR for seamless data exchange in Healthcare. *Darpan International Research Analysis*, 12(3), 403-419.
  31. Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: Smaller, faster, cheaper and lighter. *arXiv:1910.01108*.
  32. Sheng, Z., Chen, Z., Gu, S., Huang, H., Gu, G., & Huang, J. (2024). LLMs in software security: A survey of vulnerability detection techniques and insights. *arXiv:2502.07049*.
  33. Silva, P., Gonçalves, J., Antunes, N., & Vieira, M. (2024). Security and privacy of technologies in health information systems. *Computers*, 13(2), 41.
  34. Silvestri, S., Islam, S., Amelin, D., Weiler, G., Papastergiou, S., & Ciampi, M. (2023). Cyber Threat Assessment and management for securing healthcare ecosystems using natural language processing. *International Journal of Information Security*, 23(1), 31-50.
  35. Sivan, R., & Zukarnain, Z. A. (2021). Security and privacy in cloud-based E-Health System. *Symmetry*, 13(5), 742.
  36. Solanki, P., Grundy, J., & Hussain, W. (2022). Operationalising Ethics in Artificial Intelligence for Healthcare: A framework for AI developers. *AI and Ethics*, 3(1), 223-240.
  37. Thabtah, F., Hammoud, S., Kamalov, F., & Gonsalves, A. (2020). Data imbalance in classification: Experimental evaluation. *Information Sciences*, 513, 429-441.
  38. Thapa, C., Jang, S. I., Ahmed, M. S., Camtepe, S., Pieprzyk, J., & Nepal, S. (2022). Transformer-based language models for software vulnerability detection. In *Proceedings of the 38th Annual Computer Security Applications Conference* (pp. 481-496).
  39. Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). GLUE: A multi-task benchmark and analysis platform for natural language understanding. *arXiv:1804.07461*.
  40. Wang, X., Hu, R., Gao, C., Wen, X., Chen, Y., & Liao, Q. (2024, April 14). ReposVul: A repository-level high-quality vulnerability dataset. Paper presented at the 472.
  41. Wang, Y., Wang, W., Joty, S., & Hoi, S. C. H. (2021). CodeT5: Identifier-aware unified pre-trained encoder-decoder models for code understanding & generation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 8696-8708).
  42. Wuyts, K., & Joosen, W. (2020). A LINDDUN-based framework for privacy threat analysis. *Computers & Security*, 94, Article 101755.
  43. Zhou, Y., Liu, S., Siow, J., Du, X., & Liu, Y. (2019). Devign: Effective vulnerability identification by learning comprehensive program semantics via graph neural networks. In *Advances in Neural Information Processing Systems*, 32 (pp. 10197-10207).
  44. Zouev, E. (2020). Programming Languages for Safety-Critical Systems. In: *Software Design for Resilient Computer Systems*. Springer, Cham. [https://doi.org/10.1007/978-3-030-21244-5\\_11](https://doi.org/10.1007/978-3-030-21244-5_11)

# The Bitter Pill: Tracking and Remarketing on EU Pharmacy Websites

Zahra Moti, Kimberley Frings, Christine Utz, Frederik Zuiderveen Borgesius,  
and Gunes Acar

Radboud University

**Abstract.** We investigate online tracking and remarketing practices on 50 pharmacy websites in five European countries, focusing on information shared with third parties. By manually shopping for pregnancy tests and automatically analyzing the HTTP traffic data captured in HAR files, we find that users' personal data and shopping activities are routinely collected by third parties. Many pharmacy websites share product names, email addresses and phone numbers with third parties even when consent was declined. Investigating novel forms of online tracking, we find several cases of server-side tagging and CNAME-based tracking, which can be used to circumvent tracking protections offered by adblockers and modern browsers. Monitoring the advertisements targeted to our shopping profiles on several news websites and large online platform apps, we find re-targeted advertisements of the pregnancy tests we had shopped for. We further find that while declining consent reduces third-party data sharing, it does not eliminate it, and deceptive designs often discourage users from opting out. Through GDPR data access requests we reveal that companies vary in the completeness of the personal data they disclose, with none providing a full list. Overall, our study reveals widespread potential legal violations and adoption of evasive tracking technologies on websites that handle users' most sensitive personal data.

**Keywords:** Privacy, online tracking, pharmacy, online advertising

## 1 Introduction

Over the past few years, the online pharmacy sector has grown significantly, driven by the convenience of home delivery, price comparisons, and customer reviews. Online pharmacies offer convenience but pose risks due to extensive data sharing with third parties for ads and analytics. The online pharmacy section of Walgreens has also previously been shown to leak prescription information to *session replay* companies [1]. A recent investigation by The Markup showed that third-party data collection was taking place on 49 out of 50 US telehealth websites, in certain cases for targeted advertising purposes [22]. The US Federal Trade Commission investigated ad-related data sharing by GoodRx, BetterHelp, and Cerebral, resulting in multimillion-dollar settlements and bans on sharing data with advertisers like Facebook [23–25]. While these investigations showed

the risks for the US users, it is unclear whether European online pharmacy users are protected by stricter privacy laws. Our study conducts an empirical investigation to answer this question, considering novel tracking mechanisms and re-targeted advertisements.

In the context of online pharmacies, data collected by third parties may include pages visited, products browsed, purchases made, and even personal information entered during checkout. Such data can be used for relatively innocuous purposes, such as improving user experience. However, users’ activities on pharmacy websites could also be used for advertising and marketing. Previous work has shown that many telehealth websites leak personal data to third parties [22], which can reveal intimate details about a user’s private life. In regions where reproductive healthcare is contentious, tracking data may even be used in legal prosecution [6, 36].

This paper investigates the prevalence and nature of online tracking and ad retargeting (remarketing) practices on 50 European online pharmacies. Specifically, we focus on pharmacy websites that offer non-prescription medications in the four most populous EU countries —Germany, France, Spain, and Italy—as well as in the Netherlands. We focus on the most popular pharmacy websites in each country, as they attract the majority of users and reveal the tracking practices most consumers are likely to encounter. We examine the prevalence of third-party data collection by simulating a user shopping for pregnancy products. In addition, we study novel tracking mechanisms such as CNAME-based tracking and Server-Side Tagging/Tracking (SST), which bypass tracking protections that rely on blocklists. Further, we attempt to trace how the collected data is used by examining the advertisements we receive on the Web after our pharmacy browsing sessions. We also use GDPR rights to request our data from the large third parties that collect data through pharmacy websites. To evaluate the effectiveness of user controls, we compare tracking and advertising practices in two scenarios: when website visitors accept cookies and when they reject them. Overall, our contributions include the following:

- We compare tracking practices on 50 pharmacy websites across five European countries, including novel methods such as SST and CNAME-based tracking.
- We quantify personal information and product name leaks to tracker domains, showing the extent of leaks even if the user declines consent.
- Through an exploratory study of retargeted ads based on our shopping activity on pharmacy websites, we show that even sensitive products such as pregnancy tests are used for ad retargeting.
- We compare client-side data collection by major platforms with their GDPR data access responses, revealing significant discrepancies.

## 2 Related Work

Our study builds upon prior work on web tracking in health-related contexts and considers novel tracking techniques.

**Tracking on health-related websites.** Several papers investigated third-party tracking on health-related websites, with most focusing on the United States. Friedman et al. [28] researched abortion clinic websites, while McCoy et al. [38] focused on websites related to COVID-19. Both studies relied on webXray [54] to log third-party requests and cookies, a scope that likely underestimates harder-to-detect methods such as server-side tracking. Nevertheless, 99% of pages in both studies contained third-party trackers. In 2022, The Markup collaborated with STAT to investigate telehealth websites in the US [22]. They analyzed the presence of third-party trackers and shared data type (e.g., product details or shopping cart items). Among 50 telehealth websites, all but one sent personal details—often hashed or even plaintext email addresses—to major tech companies, most during checkout or questionnaire submission. In 2023, a study of 12 U.S. drugstores [53] found that all shared information about viewed or purchased products with major tracking companies.

Research into the tracking practices of European health-related websites is more sparse. Rauti et al. [49] analyzed the tracking practices on 163 Finnish online pharmacies. They found that 57 (35%) pharmacies leaked both the queried prescription name and identifying personal data. Yu et al. [55] studied 19,483 hospital websites in 152 countries—including 5,936 in Europe—and found tracking scripts on 53.5% of sites worldwide (48.8% in Europe) and tracking cookies on 14.6% (7.5% in Europe). Cookiebot, a Danish company, conducted similar research on EU health and government websites and found that 52% of EU public health service websites contained commercial trackers [8].

**Emerging web tracking techniques.** As major browsers block third-party trackers and cookies, websites increasingly adopt new techniques to bypass these restrictions. One such method is CNAME-based tracking, which uses DNS aliases to disguise trackers as first-party resources. Dimova et al. [14] presented a large-scale, longitudinal study of this technique, finding increasing adoption, especially on high-traffic sites, and posing serious security risks due to bypassing the Same-Origin Policy. Another emerging technique is Server-Side Tagging (SST), introduced by Google in 2020 [26]. Unlike client-side tracking, SST shifts data collection to a server, hiding tracking activity from the user’s browser. In a recent study, Fouad et al. [27] investigated SST at scale. They flagged SST domains by identifying subdomains absent in pre-2020 crawls, confirming they were registered to entities other than the parent domain and that their requests included tracking data previously sent elsewhere.

**Our approach.** Unlike prior work, such as Rauti et al. [49], we examine tracking after consent is declined, quantify email and phone number leaks, identify CNAME cloaking and server-side tracking using a history-free detector, and link these leaks to retargeted ads seen on the Web and mobile. To identify SST endpoints on websites, we took a different approach from Fouad et al. [27], who compared website behavior before and after SST implementation and found SST on 28 websites. Instead, our analysis relies on fixed URL parameter structures and request initiators, yielding a much higher prevalence of SST. A caveat is that our method focuses on Google Tag Manager’s SST implementation due to its



popularity, rather than detecting generic server-side tracking. Finally, we leverage GDPR data access rights to compare data collected on pharmacy websites by large online platforms to data disclosed in response to subject access requests.

### 3 Methods

We investigate tracking and advertising practices on 50 pharmacy websites across five EU countries. We simulated shopping for pregnancy tests under two consent conditions (accept/reject), using fresh browser profiles, predefined personas, and VPNs. We analyzed HTTP traffic to identify trackers and detect techniques like server-side tagging and CNAME cloaking. To assess advertising, we monitored targeted ads on news websites and mobile apps. Finally, we compared GDPR data access responses with our observed tracking activity.

#### 3.1 Website Selection

When studying online pharmacies, we distinguish between those offering prescription and non-prescription medications. As regulations differ across countries, we focus exclusively on websites selling non-prescription drugs to maintain consistency. We target popular, legitimate pharmacy websites, as users are more likely to visit them. Under Directive 2011/62/EU [20], legitimate pharmacies must register with national authorities and link to an official database. We retrieved registered pharmacies from Germany [7], France [47], Italy [42], Spain [4], and the Netherlands [41]. Popularity rankings were based on Similarweb’s DigitalRank [50]. We selected the top ten pharmacies per country to balance breadth and manual feasibility, while ensuring our sample includes the sites most online shoppers for pharmacy products are likely to visit.

#### 3.2 Data Collection

We collected data in two distinct phases, as described below. The first phase focused on tracking and web-based retargeting (Algorithm 1), while the second focused on ads on large online platforms’ mobile apps (Algorithm 2).

**Algorithm 1: Measurement of Tracking and Targeted Ads.** We followed a fixed procedure for each website to capture all relevant HTTP traffic in a reproducible manner. Algorithm 1 presents a high-level overview of our data collection process, outlining the steps we followed to capture HTTP traffic across various consent modes, countries, and pharmacy websites. We started with a fresh browser profile for each website and followed the steps below for each consent mode in every country, across all pharmacy websites in our dataset:

1. Open Developer Tools, enable HTTP logging, and detach the panel to avoid detection influencing tracking behavior [44].
2. Load the homepage and handle the cookie dialog per consent mode.

**Algorithm 1** Tracking and Targeted Web Ads Analysis

```

1: for each consent mode do
2:   Prepare predefined personal info
3:   for each country do
4:     Use VPN to simulate location
5:     for each pharmacy website do
6:       Create a fresh profile
7:       Checkout a product
8:       Save the HAR file
9:       Check news websites for
        ads
10:    end for
11:  end for
12: end for

```

**Algorithm 2** Analysis of Data Collection by Large Online Platforms

```

1: for each consent mode do
2:   Create a fresh profile
3:   Log in to platform accounts
4:   for each country do
5:     Use VPN to simulate location
6:     for each pharmacy website do
7:       Checkout a product
8:     end for
9:   end for
10:  Check platform apps for ads
11:  Scroll for 2 minutes
12:  Wait until the next day
13: end for

```

3. Search for pregnancy tests or browse the menu if no results appear.
4. View the first product page, return to the results, and open the next product.
5. Add the product to the cart, adjusting quantity if required.
6. Proceed through checkout as far as possible without placing the order, using guest checkout and predefined personal data; register if required.
7. Save all HTTP requests and responses as an HTTP Archive (HAR) file.

We leveraged HAR files to identify tracking-related requests using the uBlock Origin Core npm package [33]. We relied on uBlock Origin’s default filter lists, including EasyList and EasyPrivacy, among others [32]. We then mapped tracker domains to their respective owner entities using DuckDuckGo’s entity map [16].

**Targeted Ads on the Web.** After visiting each pharmacy website, we visited a set of news websites to observe any targeted or retargeted ads resulting from the prior shopping activity. We used Similarweb [50] to select the top five “content publishing” sites per country and five global sites, as these categories include ad-supported news websites and align with prior ad targeting research [13]. We excluded duplicates and subscription-based, ad-free sites. To analyze ad behavior and disclosures, we followed these steps:

1. Visit the homepage and interact with the cookie banner.
2. Scroll to the bottom of the page or stop after 10 seconds for infinite scrolling.
3. If pregnancy ads appear, click the AdChoices icon for the explanation page.
4. Visit two inner pages (prioritize the most prominent items and avoid health-related pages) and follow steps 4 and 5 above.

We acknowledge that our data collection incurred some cost on the pharmacy websites’ advertising budgets by causing ad impressions during the advertisement monitoring. We believe the societal benefits of our investigation outweigh its negligible cost to advertisers.

**Algorithm 2: Data Collection by Large Online Platforms.** To investigate whether data collected by third parties was used for personalized ads and disclosed to users properly, we created separate Instagram, Microsoft, TikTok, Facebook, Snapchat and Google accounts for each consent mode (accept/reject) on two iPhones. Algorithm 2 outlines the data collection process: For each consent mode, we created a fresh browser profile, logged into the six platform accounts, and searched for pregnancy tests on each pharmacy site, proceeding through checkout as far as possible without payment. After completing daily website visits, we monitored the online platforms’ mobile apps three times a day in two-minute scrolling sessions, continuing for up to a week<sup>1</sup>. We captured screenshots of any ads related to health, pharmacies, or pregnancy tests. Finally, we requested and examined data downloads from these platforms (§3.4).

### 3.3 Measurement Setup

Our experiments were conducted using Chromium browsers running on Ubuntu 24.04.1 LTS. For sites with a cookie banner, we collected data in both “accept” and “reject” modes; if no banner appeared, the same data was used for both. We used two separate computers per mode to minimize the cross-contamination risk between different consent modes. Visiting the same website twice, even after clearing cookies and browser history, could still allow tracking through fingerprinting, potentially influencing ads based on prior visits. Using two computers minimizes the risk of cross-contamination between browsing sessions of different consent modes. We used a predefined persona on each computer during checkout, allowing us to later check if personal data was leaked to third parties. To access the websites from their respective countries of origin, we used Mullvad VPN [43]. This enabled us to better impersonate a local pharmacy shopper, which may be relevant for ad targeting.

### 3.4 Detecting Tracking Methods and Leaks

**CNAME-based Tracking.** A potential method to bypass blocklist-based tracking protection is CNAME-based tracking. To evade blocking, the website owner maps a first-party subdomain to the tracker’s domain via CNAME records. Due to the increasing popularity of this technique [14], many defenses, such as uBlock Origin and AdGuard—have introduced countermeasures [31, 39]. uBlock Origin, for instance, performs DNS lookups and replays filtering with the resolved CNAME address. We adopt this method, which is enabled by uBlock Origin’s `cnameReplayFullURL` option. If a hostname has a CNAME record, we replace it with the resolved domain and rerun tracker detection using the uBO Core npm package [33]. DNS lookups are automated using the `dnspython` library [15].

<sup>1</sup> We did not monitor ads on Google mobile apps, as our Web-focused measurement (Algorithm 1) targets Google ads on websites. For Microsoft, we used the Bing app; for others, we used their respective mobile apps.

**Server Side Tagging.** Many websites embed multiple third-party resources, adding performance overhead due to increased page weight. Adblockers and tracking protections now offered by many mainstream browsers block tracking- and advertising-related third-party traffic. Server Side Tagging (SST) was proposed as a way to reduce this overhead of third parties while also bypassing tracking protections [26]. In SST, the end user’s browser or mobile app only sends tracking and analytics data to a single server, which then relays it to multiple third parties (a.k.a. tags). SST may make it challenging to identify the third parties collecting data on a website, and hence poses a transparency problem. To detect SST usage, we relied on a simple observation. Despite the change in endpoints, many URL parameters used to send data remain the same. For instance, in both SST and non-SST integrations, Google Analytics uses the parameters `dt`, `dl`, and `sr`, which correspond to page title, page URL, and screen dimensions, respectively. However, instead of manually picking parameters, we automated the parameter detection using our dataset to bootstrap the process. We first identified all requests triggered by Google Tag Manager (GTM) scripts using `initiator` fields, since SST uses GTM under the hood [29]. To detect self-hosted GTM scripts, we used a pattern we extracted from the official GTM scripts. We then took the intersection of URL parameters observed in requests triggered by GTM scripts. This yielded a list of 36 parameters, which we searched for in all requests. Similar to Fouad et al. [27], we then verified whether these requests were indeed SST by retrieving the IP address pointing to the first-party subdomain in the request and checking to which organization this IP address is registered. Then we used the terminal command `whois` to check whether the first-party subdomain organization differs from the website. We also used the request initiators for further confirmation.

**Detecting Product Name and Personal Information Leaks.** When placing an order, users provide personal information such as name, address, email, and product details, which may be shared with third parties. Identifying when and how different types of data are shared can be challenging, particularly across languages. To enable systematic analysis, we compiled search terms including product names and persona details used during checkout. Personal information or product names can be sent to tracking parties using encodings or cryptographic hashes such as SHA-256 [40]. To detect such transformed leaks, we followed Englehardt et al. [17] to search for permutations of various encodings and hashes (e.g., Base64, SHA-256) in request URLs and POST bodies.

**Data Retrieval from Third Parties.** We retrieved personal data from major platforms via their account settings or privacy centers. From Google, we exported service-wide activity data. Facebook and Instagram provided lists of companies sharing off-site activity with Meta, including browsing and purchases [35]. TikTok’s “Ads and data” section contained advertising-related data. Microsoft’s Privacy Dashboard included ad profiles and inferred interests. From Snapchat, we downloaded user data such as purchase history, memories, and other account activity. Note that data requests were made using automated tools provided by the platforms, without contacting any employees.

Table 1: Most common categories of third-party entities found on pharmacy websites. It shows the number of websites where requests to these domains observed, along with the number of distinct request domains and entities per category.

Entity Category	Websites		Request domains		Request entities	
	Accept	Reject	Accept	Reject	Accept	Reject
Advertising	50	49	73	52	58	41
Ad-motivated tracking	50	49	72	50	56	38
Analytics	50	45	45	32	37	27
3rd party analytics marketing	49	45	34	25	33	24
Audience measurement	49	43	27	20	24	17
Ad fraud	39	28	8	7	5	4

### 3.5 Analysis of Consent Notices

To provide insights into the mechanisms that online pharmacies offer customers to control the processing of their personal data, we manually inspected screenshots taken from each pharmacy’s main page for the presence of consent notices and the options they offer. We focus on control mechanisms available on the first layer of the notices, as only a few people are willing to explore deeper layers of consent notices for options to deny consent [46]. Our analysis was guided by the requirements of European data protection authorities that it must be as easy to reject data collection as to consent to it [19]. Thus, we annotated the screenshots of consent notices for the interaction options offered to website visitors on the first layer and their formatting and placement within the banner. One of the authors did the annotations, and edge cases were resolved in joint discussion.

## 4 Findings

### 4.1 Third Parties and Tracking

We analyzed HTTP requests and responses from the captured HAR files to identify third parties and various types of data sharing with them. All pharmacy websites embedded at least one third-party domain, regardless of giving or declining consent. The median number of third-party domains per site varied substantially across countries—16 in France and 56 in Italy—with Italian and German sites embedding the most (Figure 1). Rejecting cookies reduces the number of third-party domains across all countries, with Germany seeing the largest drop. Also, we identified a substantial number of pharmacy websites where third parties set cookies with the `SameSite=None` attribute and a lifespan exceeding two months—47 and 33 websites in accept and reject mode, respectively. Analyzing cookie purposes is out of the study scope, but `SameSite=None` cookies enable third parties to track users across domains.

**Tracker entities.** A large portion of third-party embeds on pharmacy websites were classified as trackers. Figure 1 shows the median number of tracking entities per website, revealing substantial variation across countries: French

websites had the fewest entities (median of nine in accept mode), while Italian websites had the most (median of 43.5). Rejecting cookies generally reduced the number of trackers, except for French sites.

**Most prevalent trackers.** As shown in Table 2, Google appeared on 96% of websites, followed by Microsoft, Facebook, and PayPal. Another frequently encountered third party is Criteo, which specializes in personalized advertising [12]. Other prevalent trackers such as Awin [5], Outbrain [48], and Taboola [52] were linked to marketing, native ads, and content recommendations. We also identified ID5 [34], a provider of privacy-focused identity solutions designed to replace third-party cookies. The presence of these trackers on pharmacy sites raises privacy concerns, as users may not expect health-related browsing activities to be used for ads or data sharing.

Table 2: Frequent third-party entities on pharmacy websites.

Third-party entity	Accept	Reject
Google	50	48
Microsoft	36	16
Facebook	32	13
Virtual Minds	18	7
Criteo	16	3
ID5	16	2
PayPal	15	15
Trusted Shops	14	12
Awin	13	4
Outbrain	13	2

**Prevalence of third-party categories.** Table 1 summarizes third-party service categories across pharmacy sites. Domains were categorized based on the Tracker Radar dataset [16]; with some domains belonging to multiple categories. “Advertising” and “Ad motivated tracking” appeared on nearly all websites, with over 70 unique domains and 50 entities focused on tracking.

#### CNAME-based Tracking.

To detect CNAME-based tracking, we replaced request hostnames with their CNAME records and reran detection using uBO Core (§3.4). Focusing on requests that were detected as trackers only after the CNAME replacement, we identified six distinct pharmacy websites that use CNAME-based tracking (Table 3). Registrable domains of all CNAME records appear in the EasyPrivacy list blocklist. Etracker.com describes how site owners can “avoid data loss due to ad blocking” using CNAME records [18]. Similarly, Mapp’s help pages [37] explain how to set up first-party tracking by defining a CNAME record pointing to `go-direct.flx1.com`, a domain used by two pharmacies. In another case, despite rejecting consent, our persona’s name and email were sent to Spotler (`activate.deonlinedrogist.nl`), which provides email marketing services [51].

**Server Side Tagging.** Using the URL parameters in GTM traffic (§3.4), we found that 19 of the 50 sites used SST. In all cases, a first-party subdomain

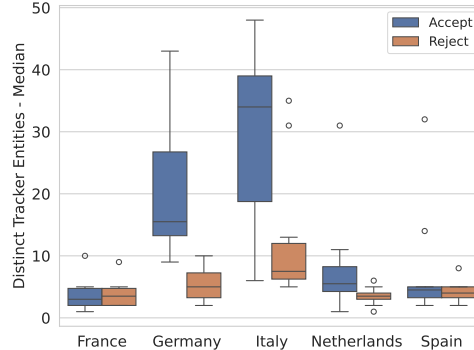


Fig. 1: Median of tracker entities per site by consent mode and country.

Table 3: Detected CNAME-based tracking domains, showing the original request host, the resolved CNAME and the consent mode(s) in which they were observed.

Loc.	Website	Request host	CNAME	Cons.
DE	medikamente-per-klick.de	e.medikamente-per-klick.de	customer.etracker.com	Both
IT	farmasave.it	ddbm2.paypal.com	ddbm2.paypal.com.[...].datadome.co	Reject
IT	topfarmacia.it	dmp.email.topfarmacia.it	go-direct.flx1.com	Both
IT	docpeter.it	dmp.mapp.docpeter.it	go-direct.flx1.com	Both
IT	1000farmacie.it	the.sciencebehindecommerce.com	tag.device9.com	Accept
NL	deonlinedrogist.nl	activate.deonlinedrogist.nl	ujemkxutgo.relay.squeezely.tech	Both

of the pharmacy website was used. In 14/19 cases, the SST endpoint was used in both accept and reject modes. Four of the five German websites used the SST endpoint only in accept mode, while the only French pharmacy used it only in reject mode. We found that 12 of the 19 SST servers were hosted on Google, easing the setup of SST servers [29]. To determine the hosting details, we used a combination of **Via** and **Server** response headers captured in the HAR files and additional WHOIS information we queried for the server IP addresses. The majority of SST endpoints used the default `/collect` path of Google Analytics, while two Italian pharmacies used a random path starting with **ngt**. Use of a random path could be an additional effort to evade blocking.

## 4.2 Product Name and Personal Information Leaks

We examined two types of information leakage: product names and identifying personal details. To prevent false positives, we only considered leaks to third-party domains and to 19 SST hostnames identified in §4.1.

### Product name leaks.

In accept mode, 34 websites leaked the product name, 77% via URLs and 23% via POST request bodies. 28 websites leaked product names even when consent was declined. Google was the top recipient of product name leaks (36 Accept, 23 Reject; Table 5), followed by Microsoft, ByteDance, and Facebook.

While leaks to **doubleclick.net** dropped substantially in reject mode (24 to 3), leaks to **google-analytics.com** increased (12 to 17), which may be a fallback domain in reject mode. Product names are still leaked to several third

Table 4: SST endpoints by country (Loc.) and used consent mode (A: Accept, R: Reject).

Loc.	SST Endpoint	Google Hosted	Cons.
DE	measure.medpex.de/g/collect	True	A
DE	tmsst.aponeo.de/g/collect	True	A
DE	klpoz.shop-apotheke.com/g/collect	True	A/R
DE	sgtm.mycare.de/g/collect	False	A
DE	measure.docmorris.de/g/collect	True	A
IT	otasf.redcare.it/g/collect	True	A/R
IT	gtm.efarma.com/g/collect	False	A/R
IT	sgtm.farmasave.it/ngtwxyzwjg	False	A/R
IT	sgtm.docpeter.it/ngtmapwbued	False	A/R
ES	datos.farmaciasdirect.es/g/collect	True	A/R
NL	pipeline.drogist.nl/g/collect	True	A/R
NL	metrics.deonlinedrogist.nl/g/collect	True	A/R
NL	sgtm.plein.nl/g/collect	True	A/R
NL	ecom-data.trekpleister.nl/g/collect	True	A/R
NL	ecom-data.kruidvat.nl/g/collect	True	A/R
NL	inc.da.nl/g/collect	False	A/R
NL	v3-pixal-web.etos.nl/g/collect	False	A/R
NL	sst.koopjesdrogisterij.nl/g/collect	False	A/R
FR	care.soin-et-nature.com/g/collect	True	R



parties in reject mode. For instance, `efarma.com` (IT) leaked product names to six domains. In contrast, seven of ten German sites avoided such leaks, while leakage patterns in other countries remained largely unchanged (Table 6). On all sites but two, URL encoding is used when leaking the product name to third parties or SST hostnames. On `shop-apotheke.com` (DE) and `redcare.it` (IT) the product name was leaked in Base64 encoded form to `adtriba.com`, a digital marketing company [3].

**Personal information leaks.** To examine personal data leaks, we focused on email addresses and phone numbers, which uniquely identify users. As with product name analysis, we considered only third-party domains and SST hostnames. In accept mode, emails leaked in 15 cases and phone numbers in three; in reject mode, email leaks slightly dropped to 13, while phone leaks rose to four. SHA-256 was the most common hashing/encoding method observed in email leaks (39 of 164). Overall, hashed email leaks were detected on five distinct sites. Facebook received hashed emails from three websites in accept mode and from two sites in reject mode (`boticas23.com`, `okfarma.es`). Other domains receiving hashed emails include `awin1.com`, `zenaps.com`, `dynamicyield.com`, `pinterest.com` and `tiktok.com`. Notably, `awin1.com` received a salted hash, which prevents linking user identities via hashed emails.

### 4.3 Consent Notices

All but one of the 50 online pharmacies (`pharmaciedesdrakkars.com`, France) displayed a consent notice. Since consent notices often employ deceptive design patterns to steer website visitors towards accepting all cookies and tracking technologies [46], our analysis focused on whether the pharmacies transparently communicated options to decline such data collection. As we will discuss in §5, EU privacy law requires consent to be “freely given,” which requires equal prominence for “Accept” and “Reject” options. A 2023 report by the European Data Protection Board found that the majority of surveyed national data protection authorities considered embedding refusal links within text paragraphs invalid unless they are visually highlighted to attract users’ attention [19]. We only found five pharmacies (two from each NL and ES and one from IT) that display “Accept” and “Reject” as equally prominent options on the first layer. While 27 additional pharmacies did feature a “Reject” option on the first layer, 21 used color highlighting to point visitors towards the “Accept” option and nine did not place the “Reject” option next to the “Accept” button. 12 pharmacies did not feature any explicit “Reject” option on the first layer, but four of them had an “Accept

Table 5: Number of sites leaking product names to third-party entities.

Entity	Accept	Reject
Google	36	23
Microsoft	23	7
ByteDance	7	0
Facebook	4	1
Virtual Minds	4	0

Table 6: Number of sites leaking product names per country and consent mode.

Country	Accept	Reject
Germany	10	3
Spain	10	9
Italy	9	10
Netherlands	8	9
France	7	7



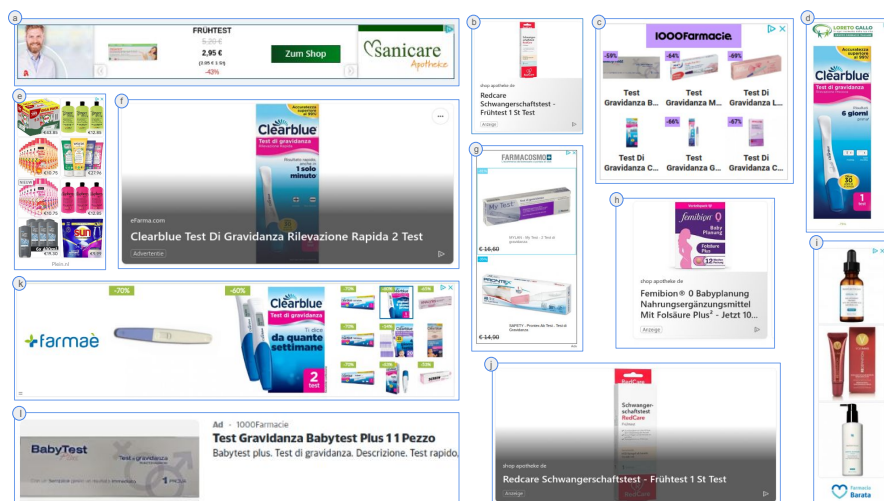


Fig. 2: Examples of targeted ads observed during the experiment.

necessary [cookies]” button instead. Thus, only five out of 50 online pharmacies featured a consent dialog that did not outright violate the requirement for “freely given” consent. A comprehensive legal analysis of whether valid consent was actually obtained would require in-depth assessment on a case-by-case basis.

#### 4.4 Advertisements

**Targeted Ads on the Web.** To assess the impact of targeted and retargeted ads, we visited news websites after browsing pharmacy sites (§3.2). The results showed notable cross-country differences in targeted and retargeted ads. We observed such ads from 15 of 50 pharmacy websites in the Netherlands, Spain, Germany, and Italy, but none from France. In the Netherlands and Spain, we saw no re-targeted ads but did receive pharmacy-related ads from visited sites ([plein.nl](http://plein.nl) and [farmaciabarata.es](http://farmaciabarata.es)) via Google Ads (Figure 2 e, i). In some cases, we saw pregnancy-related ads, though not for the exact items searched, suggesting broader behavioral targeting (Figure 2 h from [shop-apotheke.com](http://shop-apotheke.com) in Germany and d from [farmacialoreto.it](http://farmacialoreto.it) in Italy). Retargeted ads appeared on two out of ten German sites and three out of ten Italian sites, matching products we had browsed or added to our cart. These ads were served by Google, Microsoft, Criteo, as well as Taboola (Figure 2 l) and RTB House.

**Ads on Large Online Platform Apps.** A day after visiting pharmacy websites, our accept-mode Facebook feed showed numerous pregnancy- and baby-related posts and reels, but no ads. TikTok and Instagram displayed ads, yet none for pregnancy products or pharmacies. This absence may be due to fresh, low-credibility profiles and, for Facebook, the off-Facebook-activity setting—found

disabled after the study. In an earlier pilot with an author’s long-standing account, pregnancy-related Facebook ads did appear.

#### 4.5 Data Takeout from Third Parties

Under the GDPR, companies that process personal data must honor data access rights. Comparing each platform’s Takeout archive with our HAR logs—and the retargeted ads we later observed—shows that none provided a complete record.

Based on our HAR logs, 37 of the 50 sites contacted Google Analytics, but Google Takeout returned records for only 27. The Takeout data included only visited URLs, but omitted other data collected by Google Analytics, such as product names, prices, quantities, and cart actions. In contrast, HAR captures the full Analytics payloads, revealing complete product metadata and user-action events collected by Google. This gap highlights the incompleteness of Google Takeout data compared to the detailed, real-time tracking in its analytics services. TikTok’s “Off TikTok Activity” log includes events such as `InitiateCheckout`, `ViewContent`, and `AddToCart`, but provides minimal metadata—for example, checkout entries lack product details. In contrast, our HAR logs show seven sites sending product names and eight sending hashed personal data (email, phone, name) to TikTok, none of which appeared in the returned data. Instagram’s Takeout data listed advertisers that used our “activity or information”, including okfarma.es and unrelated brands such as Netflix and Paramount. The “ads and topics” folder, which logs viewed and clicked ads, contained no ads from pharmacy websites. Facebook’s “Activity Off Meta” Takeout yielded only generic privacy details and no records of pharmacy websites, despite ads related to pharmacies and pregnancy. Post-collection, we learned that off-Facebook activity ads were disabled, which may explain the absence of records. Microsoft’s ad dashboard showed new interest labels (e.g., Baby and Children) and served related ads on MSN (Figure 2), but the downloaded profile lacked the underlying data. Snapchat’s Takeout contained no data on our pharmacy visits, consistent with our client-side observations.

### 5 Legal Analysis

Here we provide a brief legal discussion—not an individual compliance assessment—of the tracking practices identified in this paper, focusing on the General Data Protection Regulation [21]. The GDPR generally applies to the tracking practices discussed in this paper because it applies when “personal data” such as cookies and other online identifiers, are used. The GDPR applies to companies (“data controllers”) based in the EU, but also to certain non-EU companies, e.g., if the company “monitors” the behavior of people in the EU (Art. 3(2)), as in online tracking. The online pharmacy and the tracking company are jointly responsible for GDPR compliance [10]. GDPR defines “special categories of personal data” that include “data concerning health or [...] a natural person’s sex life” (Art. 9(1)). The Court of Justice of the European Union (CJEU) stated that

data concerning health “must be interpreted broadly”, so if somebody orders a medical product at an online pharmacy, that fact constitutes data concerning health [9]. The use of sensitive personal data is prohibited, subject to specific exceptions such as for hospitals, which do not apply here.

The only possible legal basis for online tracking and targeted advertising is the Internet user’s “explicit consent” (Art. 9(2)) [11]. For consent to be valid, it needs to be a “freely given, specific, informed, and unambiguous indication of the data subject’s wishes by which [they], by a statement or by a clear affirmative action, signif[y] agreement to the processing of [their] personal data” (Art. 4(11)). This means that the individual must actively do something, e. g., tick a box or click a button. A company is not allowed to assume consent if someone continues to use a service or fails to opt out. As noted in §4.3, we saw tracking for targeted advertising without the individual’s “freely given” consent, a clear violation.

## 6 Limitations

While our 50-site sample favors depth over breadth, covering the top ten pharmacies per country likely reflects the experience of millions of users [2]. We observed targeted ads from 15 of the 50 pharmacies. The absence of ads from other websites could be due to a lack of advertising campaigns targeting the products we shopped for. Our use of fresh profiles on separate devices minimized the risk of prior browsing history influencing tracking behavior and ad delivery, though residual effects beyond our control cannot be entirely excluded. While our study focuses on pregnancy tests and results may not fully generalize to other sensitive health-related products, the observed tracking and ad targeting demonstrate how even sensitive product purchases are monitored for ad retargeting. Future investigations could also examine whether browsing for such products triggers ads for related categories, for example, baby items, to shed light on the broader profiling strategies employed by advertisers. We used manual checkouts to avoid bot detection and ensure ecological validity. Future work could explore LLM-guided automation [45], though this may trigger bot detection or ad fraud defenses. Google allows users to limit ads about sensitive topics such as “pregnancy and parenting” [30]. Due to scope limitations, we could not evaluate the effect of this opt-in setting. While we searched for various types of encodings and hashes to identify leaked data, custom encodings or obfuscation can bypass our detector. Hence, our ad targeting results should be taken as lower bounds. Our SST endpoint identification method focused on the server-side use of Google Tag Manager, rather than generic server-side tracking. Since our method relies on common URL parameters extracted from the data we collected, it may not generalize to other datasets or more customized uses of SST.

## 7 Conclusion

Users may expect a high level of privacy when shopping for health-related products online. Our findings show that even shopping for sensitive products such as pregnancy tests on most popular European pharmacy websites is subject to extensive third-party tracking for advertising purposes. Through a lightweight detection method, we identify a sharp increase in the use of server-side tracking, along with continued use of other stealthy techniques such as CNAME cloaking. Tracking often occurs without valid consent as many websites do not use compliant consent dialogs, and some ignore user choices altogether. Moreover, data access requests often yield incomplete information, leaving users in the dark about what online activities are monitored. Our findings raise significant concerns regarding transparency, user rights, and compliance with regulations.

## References

1. Acar, G., Englehardt, S., Narayanan, A.: No boundaries: data exfiltration by third parties embedded on web pages. *Proceedings on Privacy Enhancing Technologies*. p. 220–238 (2020). <https://doi.org/10.2478/popets-2020-0070>
2. Adamic, L.A., Huberman, B.A.: Zipf’s law and the internet. *Glottometrics* **3**(1), 143–150 (2002)
3. AdTriba GmbH: Future-proof marketing measurement & optimization (Oct 2024), <https://www.adtriba.com>
4. Agencia Española de Medicamentos Productos Sanitarios: Listado de farmacias que realizan la venta a distancia (2024), <https://distafarma.aemps.es/farmacom/faces/inicio.xhtml>
5. AWIN Inc.: Join our global affiliate platform (2024), <https://www.awin.com/>
6. Baker-White, E.: Facebook Gave Nebraska Cops A Teen’s DMs., <https://www.foxbes.com/sites/emilybaker-white/2022/08/08/facebook-abortion-teen-dms>
7. Bundesinstitut für Arzneimittel und Medizinprodukte: Versandhandelsregister (Oct 2024), <https://versandhandel.dimdi.de/pdfs/vhr-apo.pdf>
8. Cookiebot: Ad Tech Surveillance on the Public Sector Web (2019), <https://www.cookiebot.com/media/1121/cookiebot-report-2019-medium-size.pdf>
9. Court of Justice of the EU: Judgment in Case C-21/23, 4 Oct. 2024, <https://curia.europa.eu/juris/liste.jsf?num=C-21/23>
10. Court of Justice of the EU: Judgment in Case C-40/17, 29 July. 2019, <https://curia.europa.eu/juris/liste.jsf?num=C-40/17>
11. Court of Justice of the EU: Judgment in Case C-446/21, 4 Oct. 2024, <https://curia.europa.eu/juris/liste.jsf?num=C-446/21>
12. Criteo: The Commerce Media Platform for the Open Internet (2024), <https://www.criteo.com/>
13. Datta, A., Tschanz, M.C., Datta, A.: Automated Experiments on Ad Privacy Settings – A Tale of Opacity, Choice, and Discrimination. *Proceedings on Privacy Enhancing Technologies*. (1), 92–112 (2015)
14. Dimova, Y., Acar, G., Olejnik, L., Joosen, W., Van Goethem, T.: The CNAME of the Game: Large-scale Analysis of DNS-based Tracking Evasion. *Privacy Enhancing Technologies*. pp. 394–412 (2021)

15. Dnspython Contributors: dnspython (2024), <https://dnspython.readthedocs.io/en/latest>
16. DuckDuckGo: Tracker Radar (2024), <https://github.com/duckduckgo/tracker-radar/>
17. Englehardt, S., Han, J., Narayanan, A.: I never signed up for this! Privacy implications of email tracking. *Proceedings on Privacy Enhancing Technologies*. pp. 109–126 (2018)
18. eTracker GmbH: Set up your own tracking domain (2025), <https://help.etracker.com/en/article/set-up-your-own-tracking-domain>
19. European Data Protection Board: Report of the work undertaken by Cookie Banner Taskforce. Tech. rep. (2023), <https://www.edpb.europa.eu/our-work-tools/our-documents/other/report-work-undertaken-cookie-banner-taskforce>
20. European Parliament and the Council of the EU: Directive 2011/62/EU (2011), <http://data.europa.eu/eli/dir/2011/62/oj>
21. European Parliament and the Council of the EU: Regulation (EU) 2016/679 (2016), <https://eur-lex.europa.eu/eli/reg/2016/679/oj>
22. Feathers, T., Palmer, K., Fondrie-Teitler, S.: Dozens of Telehealth Startups Sent Sensitive Health Information to Big Tech Companies (2024), <https://themarkup.org/pixel-hunt/2022/12/13/out-of-control-dozens-of-telehealth-startups-sent-sensitive-health-information-to-big-tech-companies>
23. Federal Trade Commission: Enforcement Action to Bar GoodRx (2023), <https://www.ftc.gov/news-events/news/press-releases/2023/02/ftc-enforcement-action-bar-goodrx-sharing-consumers-sensitive-health-info-advertising>
24. Federal Trade Commission: FTC Order Prohibits Telehealth Firm Cerebral from Using Sensitive Data for Ads (2024), <https://www.ftc.gov/news-events/news/press-releases/2024/04/proposed-ftc-order-will-prohibit-telehealth-firm-cerebral-using-or-disclosing-sensitive-data>
25. Federal Trade Commission: FTC to Ban BetterHelp from Revealing Consumers’ Data (2024), <https://www.ftc.gov/news-events/news/press-releases/2023/03/ftc-ban-betterhelp-revealing-consumers-data-including-sensitive-mental-health-information-facebook>
26. Fisher, B.: Improve performance and security with Server-Side Tagging (2023), <https://blog.google/products/marketingplatform/360/improve-performance-and-security-server-side-tagging>
27. Fouad, I., Santos, C., Laperdrix, P.: The Devil is in the Details: Detection, Measurement and Lawfulness of Server-Side Tracking on the Web. *Proceedings on Privacy Enhancing Technologies* p. 450–465 (2024)
28. Friedman, A.B., Bauer, L., Gonzales, R., McCoy, M.S.: Prevalence of Third-Party Tracking on Abortion Clinic Web Pages. *JAMA Internal Medicine* **182**(11) (2022)
29. Google: Setting up a new server container (2023), <https://developers.google.com/tag-platform/learn/sst-fundamentals/4-sst-setup-container>
30. Google: Limit ads about sensitive topics on Google (2024), <https://support.google.com/My-Ad-Center-Help/answer/12155260>
31. Hill, R.: uBlock Origin works best on Firefox, <https://github.com/gorhill/uBlock/wiki/uBlock-Origin-works-best-on-Firefox#cname-uncloaking>
32. Hill, R.: uBlock Origin – make-rulesets.js (2023), <https://github.com/gorhill/uBlock/blob/491bc87e94a503a17fd11cdee35c1f1b6fea24be/platform/mv3/make-rulesets.js#L1285-L1296>
33. Hill, R.: uBlock Origin Core (2024), <https://www.npmjs.com/package/@gorhill/ubo-core>

34. ID5: ID5 – Future-proofed user identification for Digital Advertising (2024), <https://id5.io/>
35. Instagram Help Center: Why am I seeing ads from an advertiser on Instagram?, <https://help.instagram.com/609473930427331>
36. Kaste, M.: Nebraska cops used Facebook messages to investigate an alleged illegal abortion (2022), <https://www.npr.org/2022/08/12/1117092169/nebraska-cops-used-facebook-messages-to-investigate-an-alleged-illegal-abortion>
37. Mapp: Custom Track Domain (C-Name) (2025), <https://docs.mapp.com/docs/custom-track-domain-c-name>
38. McCoy, M.S., Libert, T., Buckler, D., Grande, D.T., Friedman, A.B.: Prevalence of Third-Party Tracking on COVID-19-Related Web Pages. *Journal of the American Medical Association (JAMA)* **324**(14), 1462–1464 (2020)
39. Meshkov, A.: Gotta catch 'em all: how AdGuard scanned the entire web in search of hidden trackers (2024), <https://adguard.com/en/blog/cname-tracking.html>
40. Meta: Customer File Custom Audiences (2024), <https://developers.facebook.com/docs/marketing-api/audiences/guides/custom-audiences/#hash>
41. Ministerie van Volksgezondheid, Welzijn en Sport: Aanbiederslijst online medicijnen (2024), <https://aanbiedersmedicijnen.nl/aanbieders/aanbiederslijst>
42. Ministero della Salute: Soggetto autorizzato al commercio online di medicinali (2024), <https://www.salute.gov.it/LogoCommercioElettronico/CercaSitoEComm>
43. Mullvad VPN AB: – Free the internet (2024), <https://mullvad.net/en>
44. Musch, M., Johns, M.: U Can't Debug This: Detecting JavaScript Anti-Debugging Techniques in the Wild. In: *Proceedings of the 30th USENIX Security Symposium*. pp. 2935–2950 (2021)
45. Müller, M., Žunič, G.: Browser Use: Enable AI to control your browser (2024), <https://browser-use.com/>
46. Nouwens, M., Liccardi, I., Veale, M., Karger, D., Kagal, L.: Dark Patterns after the GDPR: Scraping Consent Pop-ups and Demonstrating their Influence. In: *Proceedings of the 2020 CHI*. pp. 1–13 (2020)
47. Ordre National des Pharmaciens: Rechercher un site de vente en ligne autorisé à vendre des médicaments – CNOP (2024), <https://www.ordre.pharmacien.fr/je-suis/patient-grand-public/rechercher-un-site-de-vente-en-ligne-autorise-a-vendre-des-medicaments?vl-region=&vl-departement=&vl-commune=&vl-site=&vl-pharmacy=&vl-incumbent=>
48. Outbrain Inc.: Drive Better Business Results (2024), <https://www.outbrain.com/>
49. Rauti, S., Carlsson, R., Mickelsson, S., Mäkilä, T., Heino, T., Pirjatanniemi, E., Leppänen, V.: Analyzing third-party data leaks on online pharmacy websites. *Health and Technology* **14**, 375–392 (2022)
50. Similarweb LTD: Unlock Digital Growth (2024), <https://www.similarweb.com/>
51. Spotler: Email marketing automation with Spotler software (2024), <https://spotler.com/solutions/use-cases/email-marketing-automation>
52. Taboola: Restricted Content, Products, Services (2025), <https://taboola.com/>
53. Tahir, D., Fondrie-Teitler, S.: Need to Get Plan B or an HIV Test Online? Facebook May Know About It (2023), <https://themarkup.org/pixel-hunt/2023/06/30/need-to-get-plan-b-or-an-hiv-test-online-facebook-may-know-about-it>
54. webXray: webXray Privacy Search Engine (2024), <https://webxray.ai/>
55. Yu, X., Samarasinghe, N., Mannan, M., Youssef, A.: Got Sick and Tracked: Privacy Analysis of Hospital Websites. In: *IEEE Euro S&P Workshops*. pp. 278–286 (2022)

# PADOME: Adaptive Privacy Assistant for the Internet of Things

Edward Rochester<sup>1</sup>[0009–0003–8774–2209] and Ken Barker<sup>1</sup>

University of Calgary, 2500 University Dr NW, Calgary, AB T2N 1N4  
`{e.rochester, ken.barker}@ucalgary.ca`

**Abstract.** As the need for privacy self-management in the Internet of Things (IoT) ecosystem grows, Privacy Assistants (PAs) have emerged as a solution for assisting users with privacy management. However, many existing PAs rely on static strategies, assume perfect knowledge of user privacy preferences, and expect complete responsiveness from users to elicitation prompts. Furthermore, they overlook IoT device behavior and information available from surrounding PAs. As such, we designed PADOME, an adaptive PA that models both the user’s privacy preferences and the behavior of the IoT device. PADOME integrates preference elicitation to better understand a user’s privacy utilities, along with opponent preference modeling and surrounding PA elicitation to reduce uncertainty about the opponent’s negotiation strategy. Designed using the DUNE framework and evaluated in the GEPARD simulation environment, PADOME demonstrates improved negotiation outcomes and higher agreement success rates.

**Keywords:** Privacy Management · Internet of Things · Privacy Assistants

## 1 Introduction

The proliferation of Internet of Things (IoT) devices has led to an unprecedented scale of personal data collection and processing, raising significant privacy concerns [6]. Privacy Assistants (PAs) have emerged as a promising approach to mitigate these concerns by (semi-)autonomously negotiating with IoT devices on behalf of users [13,1,10,19]. However, many existing PAs are designed with assumptions that fail to hold in real-world deployments. They often assume that users will always respond to elicitation requests or that IoT devices will behave predictably or cooperatively [4,9]. Such assumptions may lead to ineffectively designed PAs that can significantly degrade the IoT ecosystem’s performance and substantially reduce PA user satisfaction [14,16].

In practice, users are often unwilling or unable to respond to elicitation requests [18]. As such, PAs must balance automation with user involvement to minimize user burden and power consumption due to unanswered elicitation requests, while still maintaining trust and comprehensive privacy profiles [4,16]. Additionally, negotiation opponents, *i.e.*, IoT devices, may act strategically to

influence negotiation outcomes [2]. The negotiation environment itself is often characterized by uncertainty and incomplete knowledge (*e.g.*, unclear data flows when negotiating data sharing with a smart mall’s systems), further degrading the effectiveness of traditional PAs in real-world deployments [9].

To address these challenges, we introduce *PADOME*, a Privacy Assistant with Distributed Opponent Modeling and User Elicitation. *PADOME* is designed to operate under conditions of partial cooperation from both users and other PAs. It abandons the assumption that users will respond to elicitation prompts, instead relying on observation of prior decisions and user feedback when available. *PADOME* also prompts surrounding PAs for their model of the IoT device preferences and does not assume their responsiveness. Instead, it opportunistically elicits such models when possible without depending on their availability.

*PADOME* is designed using the DUNE framework [15], which structures PA designs along four key components: Device, User, Network, and Environment. DUNE enables systematic analysis of PA behaviors by isolating these components and supporting a plug-and-play design methodology. We also implement *PADOME* into the GEPARD simulation environment [15] to evaluate its effectiveness and compare it with other state-of-the-art PA designs.

## 2 Related Work

We limit related work to recent work in the following categories: i) automated negotiation agents in data privacy, ii) PAs for IoT privacy negotiations and management, and iii) user privacy preferences elicitation algorithms.

**i. Automated Negotiations.** Baarslag *et al.* [3] presented an automated negotiation agent that uses learned user’s privacy preferences to negotiate data-sharing permissions. While the negotiated agreements were more accurate than the baseline, the work could further explore strategies that also reduce the overall effort required from the user. This work was extended by Filipczuk *et al.* [9] to allow for partial and complete offers and introduce a variant of the user privacy preference learning approach, paving the way for more flexible negotiation mechanisms. Finally, Mohammad *et al.* [12] extended Filipczuk *et al.* work to practical use cases through partial uncertainty reduction in the utility functions, which opens the door to future work on extending these methods to scenarios with uncountable or more complex outcome spaces.

**ii. Privacy Assistants.** Cha *et al.* [5] addressed the consent issues when users access nearby IoT devices from their smartphones via the Bluetooth Low Energy (BLE) iBeacon functionality, providing valuable insights for broader IoT deployment scenarios beyond the proposed case. Das *et al.* [8] developed and deployed IoT PAs, and proposed a resource registry that can be used for advertising purposes, highlighting opportunities to refine such registries to balance usefulness with minimizing potential user fatigue from advertising. More recent work by Morel *et al.* [13] outlined high-level requirements for consent in IoT, presenting a set of technical requirements for implementing a consent framework in IoT. The authors implemented a BLE-based prototype to show the presented



framework’s real-world applicability, pointing toward future research on a holistic analysis of PA systems. Finally, Alanezi *et al.* [1] presented a PA design to address individual and group context privacy concerns, inviting future work on dynamic negotiation deadlines and adaptive preference modeling.

**iii. Elicitation Algorithms.** Baarslag and Gerding [4] proposed an *optimal algorithm* for eliciting exact DS’s preferences in a single round during negotiation, which invites future work in approaches that work effectively when users provide approximate rather than exact utility values, or when elicitation occurs over multiple rounds with minimal burden. Mohammad and Nakadai [12] addressed this by relaxing the assumption of *exact* utility value return, offering a more adaptable approach. Building on both of these works, PADOME relaxes the assumption that users will always respond to elicitation, a condition that may not always be realistic as users can miss or choose to ignore elicitation attempts.

### 3 PADOME: A Novel PA Design

In this section, we present PADOME, a novel PA design that combines the following two strategies: (i) user privacy preference elicitation, and (ii) surrounding PA elicitation. Although these mechanisms are not individually novel, their combination and the assumption that elicitation may fail presents new challenges. PADOME’s novelty lies in integrating these mechanisms and overcoming technical challenges to allow the PA to manage the complexities of dynamic negotiation and elicitation states.

#### 3.1 Setting Definition

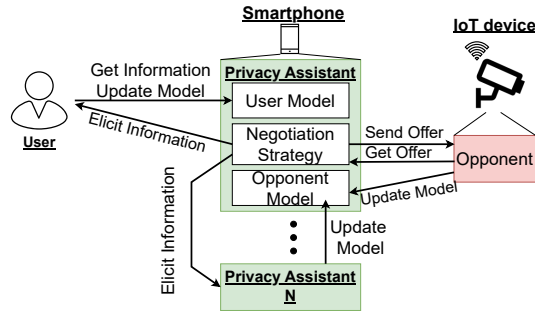


Fig. 1: Overview of interactions among the user, PAs, and IoT device.

A common assumption among PA designs is that the PA has prior knowledge of the user and opponent (*i.e.*, IoT device) models obtained from prior interactions or the population’s average. In our setting, however, the user is assumed unwilling or unable to specify their privacy preferences fully. Additionally, there

are other PAs, some having already arrived at an agreement with the IoT device. As such, for each possible offer, the PA faces the following uncertainties: (i) the opponent preferences (*opponent model*), and (ii) user's utility of an offer (*user model*). Thus, the PA can refine the opponent model by exchanging offers with the IoT device or eliciting other PAs. Broadly, the opponent model represents the IoT device's preferences and negotiation strategies. For instance, by building an opponent model, a PA may infer that the IoT device prioritizes location privacy over other PP terms and adjust its offers accordingly. Simultaneously, the PA can elicit user to refine the user model.

At each negotiation round, the PA decides whether to (i) elicit user preferences, (ii) elicit from other PAs, (iii) accept an offer, (iv) counter an offer, or (v) break off the negotiations. Each decision involves balancing benefits and costs, *e.g.*, utility at the cost of user bother. We assume *incremental* elicitation, *i.e.*, the PA can continue the elicitation process if needed. Figure 1 illustrates these interactions.

### 3.2 Formal Model

Let  $\Omega = \{\omega_1, \dots, \omega_n\}$  denote all possible offers in the negotiation, each with utility  $U(\omega)$ , which is initially uncertain. Before user preferences are elicited,  $U(\omega)$  is modeled as a stochastic variable  $x_\omega$  with cumulative distribution function  $F_\omega(x)$ , independent of other offers. The user has an exact utility function  $\tilde{U} : \Omega \rightarrow [0, 1]$  that maps all possible negotiation outcomes to a real number. This *real* utility, however, is not known to the PA. Hence, the PA maintains some probability distribution  $\hat{U}(\omega) : [0, 1] \rightarrow [0, 1]$  that represents the probability of  $\tilde{U}(\omega) = u$  for  $0 \leq u \leq 1$ , *i.e.*,  $\hat{U}(u) \equiv \text{Prob}(\tilde{U}(\omega) = u)$ . At any point in the negotiation, the PA can elicit  $\tilde{U}(\omega)$  at a cost  $c_u(\omega)$ , representing utility loss due to user bother.

The PA negotiates with the IoT device using an alternating offers protocol, modeling acceptance probability of  $\omega$  as  $p_\omega$ , calculated from prior interactions or by eliciting opponent models from other PAs at cost  $c_o(\omega)$ . The alternating offers protocol was chosen for its simplicity and widespread use in the related literature. In it, upon receiving an (counter-)offer, the PA updates the opponent model. When an offer is accepted, the negotiation ends in an agreement. The PA, however, can also choose to break off the negotiation process. Specifically, PA has a known reservation value  $r \in [0, 1]$ , which is the utility of breaking off the negotiation. Additionally, an agreement must be reached within a fixed number of exchanges  $D$ . At the end of the negotiation, the utility of the PA is given by Equation 1.

$$U = \begin{cases} U(\omega) - \sum_{\omega' \in \Omega} c_u(\omega') - \sum_{\omega'' \in \Omega} c_o(\omega'') & \text{if } \omega \in \Omega \text{ is accepted,} \\ r - \sum_{\omega' \in \Omega} c_u(\omega') - \sum_{\omega'' \in \Omega} c_o(\omega'') & \text{if no agreement reached.} \end{cases} \quad (1)$$

### 3.3 Negotiation and Elicitation Strategies

**Negotiation Strategy.** The PA aims to maximize the expected utility by calculating the expected value of different actions while considering the opponent model and expected utility if the negotiation continues.

Following Baarslag and Gerding [4], we assume the PA uses a decision function with an *aspiration value*  $\alpha_j \in [0, 1]$ , which represents the expected reward for continuing negotiation at round  $j \leq N$ , where  $N \leq D$  is the total number of rounds. Given  $\alpha_j$ , the *negotiation value* of sending an offer  $\omega \in \Omega$  is:

$$v(\omega) = p_\omega U(\omega) + (1 - p_\omega)\alpha_j, \quad (2)$$

where  $U(\omega)$  is an immediate utility payoff if the offer gets accepted with probability  $p_\omega$  and the expected future payoff of  $\alpha_j$  if the offer is rejected. We also define  $U(\omega_0) = r$ , which represents the previously defined reservation value.

At each round, if no further elicitation occurs, the optimal strategy is to select the offer with the highest negotiation value  $v^*(\Omega) = \max_{\omega \in \Omega} v(\omega)$ . We formalize this strategy in Algorithm 1.

---

**Algorithm 1:** Proposed negotiation strategy.

---

**Input:** Current negotiation state.

**Output:** One of the actions: accept, counter-offer, or break-off.

```

1 while  $j \leq D$  do
2   for  $\omega \in \Omega$  do
3     update( $p_\omega$ );
4   UserElicitation() // presented in Algorithm 2;
5   PAElicitation() // presented in Algorithm 3;
6    $\omega \leftarrow \arg \max_{\omega' \in \Omega} (p_{\omega'} U(\omega') + (1 - p_{\omega'})\alpha_j)$ ;
7   return  $\begin{cases} ACCEPT & \text{if } \omega \text{ was offered,} \\ BREAK-OFF & \text{if } \omega = \omega_0, \\ SEND(\omega) & \text{otherwise,} \end{cases}$ 

```

---

Negotiation stops if deadline  $D$  is reached, computed dynamically as:

$$D = \lfloor D_{\text{Base}} + F_M + F_{\text{Network}} + F_{\text{Distance}} + F_{\text{PP}} + 0.5 \rfloor, \quad (3)$$

where  $D_{\text{Base}}$  is the base deadline.  $F_M$  is the user factor, which accounts for the number of other users  $M$  and reflects resource burden on the ecosystem from other negotiations. The network factor  $F_{\text{Network}}$  captures differences in network technologies; *e.g.*, networks supporting small payloads may require more transmission rounds. The distance factor  $F_{\text{Distance}}$  scales with the average distance of PAs from the IoT device.  $F_{\text{PP}}$  is the privacy policy (PP) size factor.

**User Elicitation Strategy.** Using the negotiation value  $v^*(\Omega)$ , the elicitation strategy determines which offers, if any, to elicit from the user. This is a sequential decision problem, since each choice depends on the offers elicited so far and

on the probability that the user may ignore an elicitation. Therefore, the goal is to find (i) an optimal *sequence* of offers to elicit and (ii) a strategy specifying when to start and stop the elicitation.

We define the *user elicitation state* as  $\mathcal{E}_u$ , defined by  $\langle \bar{\Omega}, y \rangle$ , where  $\bar{\Omega}$  are all non-elicited offers and  $y = v^*(\Omega)$ . The goal is to formulate a user elicitation policy  $\pi_u$ , that, given  $\mathcal{E}_u$ , determines whether to elicit an offer or to proceed with negotiation. The utility of the  $\pi_u$  is:

$$U(\pi_u, \mathcal{E}_u) = \begin{cases} y & \text{if } \pi_u(\mathcal{E}_u) \notin \bar{\Omega}, \\ P_u \int_{-\infty}^{\infty} U(\pi_u, \mathcal{E}'_u) dF_{x_{\pi(\mathcal{E})}} - c(\pi_u(\mathcal{E}_u)) & \text{otherwise,} \end{cases} \quad (4)$$

where  $P_u$  is the probability of successfully eliciting the user and  $\mathcal{E}'_u = \langle \bar{\Omega} \setminus \{\pi_u(\mathcal{E}_u)\}, \max(y, v(x)) \rangle$  is the updated state after observing  $x_{\pi_u(\mathcal{E}_u)}$ .  $P_u$  can be estimated using models of user engagement [11,17].

Using Equation 4, we are looking to find  $\pi_u^* = \arg \max_{\pi_u} U(\pi_u, \mathcal{E}_u)$ . Note that if *exact* values of all offers are known,  $U(\pi_u^*, \langle \emptyset, y \rangle) = y$ . Otherwise, we calculate the negotiation value  $x_\omega^v$  of a non-elicited offer  $\omega \in \bar{\Omega}$  using the random variable  $x_\omega$  as follows:

$$x_\omega^v = p_\omega x_\omega + (1 - p_\omega) \alpha_j. \quad (5)$$

Compared to Equation 2, Equation 5 relies on a random variable instead of the *exact* utility value since the *real* value of offer  $\omega$  is unknown.

Recall that  $\pi_u$  determines whether the PA stops with utility  $y$ , or elicits an offer  $\omega \in \bar{\Omega}$  by sampling  $x_\omega$  at cost  $c(\omega)$ , while taking the following into the account:

1. If  $x_\omega > y$ , the PA has found a better offer with the expected utility  $U(\pi_u, \langle \bar{\Omega} \setminus \{\omega\}, x_\omega \rangle)$ ;
2. Otherwise, if  $x_\omega \leq y$ , nothing changes, except that PA has observed the value of  $x_\omega$ , so the new expected utility is  $U(\pi_u, \langle \bar{\Omega} \setminus \{\omega\}, y \rangle)$ .

Given the above, Equation 4 must satisfy the following recursive relation [4]:

$$U(\pi_u, \mathcal{E}_u) = \max\{y, \max_{\omega \in \bar{\Omega}} \{U(\pi_u, \langle \bar{\Omega} \setminus \{\omega\}, y \rangle) \times F_\omega^v(y) - c(\omega) + \int_{x=y}^{\infty} U(\pi_u, \langle \bar{\Omega} \setminus \{\omega\}, x \rangle) dF_\omega^v(x)\}\}, \quad (6)$$

where  $F_\omega^v(x)$  denotes the corresponding cumulative distribution function.

The relation for  $\pi_u$  in Equation 6 is a form of the Bellman equation, solvable using the index-based method presented by Baarslag and Gerding [4]. We define the index  $z_\omega^v$  as the solution to  $\int_{z_\omega^v}^{\infty} (x - z_\omega^v) dF_\omega^v(x) = c(\omega)$ .

The resulting elicitation strategy  $\pi_u$  (presented in Algorithm 2) is: *Elicit the offer  $\omega \in \bar{\Omega}$  with the highest index  $z_\omega^v$ , if it is higher than  $v^*(\Omega)$ ; update the  $v^*(\Omega)$  if the realized value is higher, and repeat. Stop the elicitation if the highest index is less than  $v^*(\Omega)$ , or when all offers are in  $\Omega$ .*

**Algorithm 2:** Proposed user elicitation strategy.

---

```

1 elicitationCost  $\leftarrow$  0;
2 for  $\omega \in \bar{\Omega}$  do
3    $z_{\omega}^v \leftarrow$  Solve  $\int_z^{\infty} (x - z) dF_{\omega}^v(x) = c(\omega)$  for  $z$ ;
4    $v \leftarrow \max_{\omega \in \Omega} (p_{\omega} U(\omega) + (1 - p_{\omega}) \alpha_j)$ ;
5    $\omega \leftarrow \arg \max_{\omega' \in \Omega} z_{\omega'}^v$ ;
6   while  $z_{\omega}^v \geq v$  or  $\bar{\Omega} \neq \emptyset$  do
7      $U(\omega) \leftarrow \text{ElicitFromUser}(\omega)$ ;
8     elicitationCost  $\leftarrow$  elicitationCost +  $c(\omega)$ ;
9      $\Omega \leftarrow \Omega \cup \{\omega\}$ ;  $\bar{\Omega} \leftarrow \bar{\Omega} \setminus \{\omega\}$ ;
10     $v \leftarrow \max(v, p_{\omega} U(\omega) + (1 - p_{\omega}) \alpha_j)$ ;
11     $\omega \leftarrow \arg \max_{\omega' \in \bar{\Omega}} z_{\omega'}^v$ ;

```

---

**PA Opponent Model Elicitation Strategy.** We use an information-theoretic approach, utilizing entropy and information gain to guide the PA elicitation strategy. Specifically, let  $\mathcal{P} = PA_1, PA_2, \dots, PA_M$  represent the set of PAs, where each  $PA_i$  is an individual PA that will send requests to the surrounding PAs. Let  $R$  denote the set of PA responses, *i.e.*, opponent models.

**Algorithm 3:** Proposed PA elicitation strategy.

---

**Input:**  $PA_i, \mathcal{P}$ , reservation value  $r$ .  
**Output:** Updated opponent model.

```

1 for  $PA \in \mathcal{P} \setminus PA_i$  do
2   Determine  $PA$  response likelihood  $p(PA)$ ;
3 for  $\omega \in \Omega_{PA_i}$  do
4    $H_0 = -\sum_{i=1}^{M-1} p_{\omega} \log_2 p_{\omega}$ 
5    $H_{\text{new}} = H_0 \times \frac{1}{M-1}$ 
6    $I = p(PA) \times (H_0 - H_{\text{new}})$ 
7    $U \leftarrow I - \frac{w_t \cdot \log(1+c(t))}{w_p \cdot \log(1+c(p))}$ 
8   if  $U > r$  then
9      $R \leftarrow \text{BroadcastToPAs}()$ ;
10     $p_{\omega_{\text{new}}} = \frac{p_{\omega} + \text{median}(R)}{2}$ 

```

---

Algorithm 3 shows the proposed PA elicitation strategy, which is designed to balance information gain with time and power consumption costs. In particular, we calculate initial and new entropy values (lines 4 and 5), provided information from  $M - 1$  PAs. We then find the expected information gain (line 6) and use it to calculate the utility (line 7). While our utility function incorporates time and power consumption, it can be extended to account for other factors. We then compare the utility against the reservation value (line 8) to determine if it

is “worth” broadcasting the request to PAs in the environment, and if yes, we update the opponent model based on the received responses (line 10).

## 4 Experiments

This section presents PADOME’s input parameter analysis and evaluation against state-of-the-art PA designs across various scenarios and network technologies.

### 4.1 Input Parameter Analysis

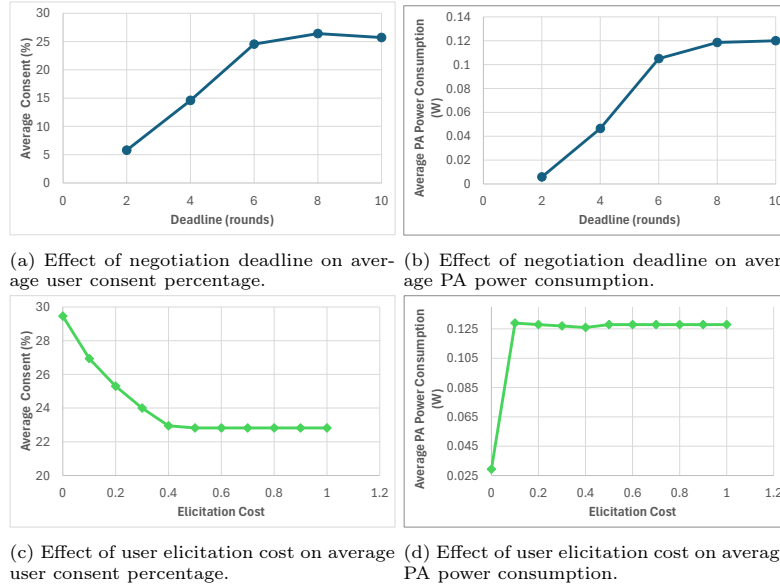


Fig. 2: Effect of input parameters on the PADOME performance.

As part of the PADOME design process, we evaluated the effect of input parameters on its performance. Specifically, we examined the impact of user elicitation cost and negotiation deadline on the *Average User Consent Percentage* and *Average PA Power Consumption*. The results are averaged over 10 times using BLE and Shopping Mall. For parameters not under investigation, we use the following fixed values: reservation value of 0.25, elicitation cost of 0.05, and base deadline of 4.

From Figures 2(a) and 2(b) we can observe that: (i) longer negotiations increase average consent percentage up to a “saturation” point, beyond which other factors, *e.g.*, network coverage, limit the PA and (ii) more negotiation rounds lead to higher power consumption until reaching a plateau. These results align with Baarslag and Gerding [4], who showed that increasing elicitation cost

reduces negotiation utility. Based on this, we set a base deadline of 4 with a dynamic increase as per Equation 3.

From Figures 2(c) and 2(d) we can observe that: (i) higher elicitation costs reduce average consent percentage, as fewer opportunities arise for the PA to gather user preferences and (ii) in the best case scenario, when PA has no cost for user elicitation, the elicitation cost has no impact on power consumption. The latter result is notable, suggesting that eliciting user preferences does not affect the negotiation flow as anticipated. Based on these results, we set an elicitation cost of 0.05.

## 4.2 Experimental Methodology

We implemented PADOME in GEPARD simulator [15]. We ran tournament-style simulations with 25 runs per design and compared them across three available scenarios: Hospital, University, and Shopping Mall. We employed BLE, ZigBee, and LoRa as network technologies and compared PADOME against three state-of-the-art PA negotiation protocols: Alanezi, Cunche, and Concession.

In our experiments, we adopted the effective communication ranges as specified in [15]: 50 m for BLE, 100 m for ZigBee, and 10,000 m for LoRa. The spatial dimensions for each scenario were also taken from the same reference, with the Hospital measuring 40 m, the University 80 m, and the Shopping Mall 120 m. These parameters are noted as they have a significant impact on the experimental results discussed in this section. For instance, BLE performance in the larger Shopping Mall scenario is expected to be poorer compared to the smaller Hospital scenario due to its limited communication range.

## 4.3 Experimental Results

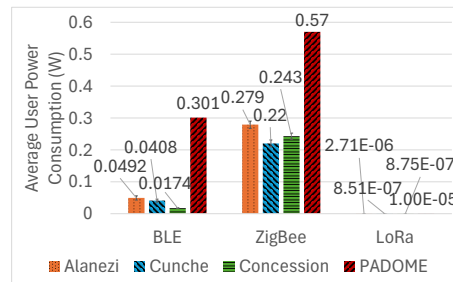


Fig. 3: Average power consumption in Hospital.

**Average User Power Consumption.** Figure 3 shows the average per-user power consumption in the Hospital scenario. We calculated the combined average user device power consumption and averaged over the total number of runs. From

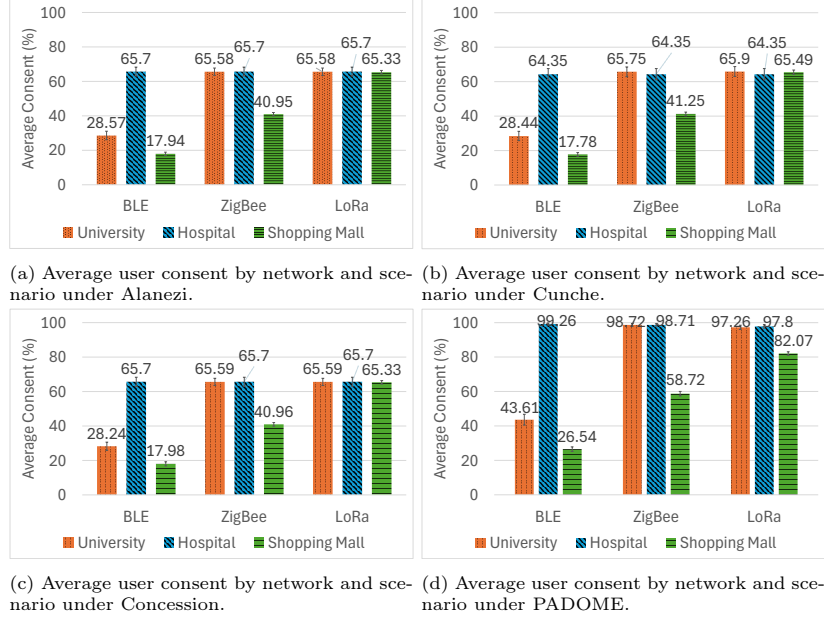


Fig. 4: Average user consent across different networks and scenarios under various network protocols.

the figure, we can observe that PADOME-based PA, due to opponent model elicitation, results in up to 17 times higher power consumption than Concession-based PA, both using BLE, which would require the IoT device to rely on the external power supply to operate.

**Average User Consent.** We measured average user consent as a percentage of consent collected and averaged across all runs. From Figure 4, we can observe that the average user consent achieved by the non-PADOME-based PAs lie within 1% of each other. This observation is notable, as it emphasizes the influence of the scenario and communication technologies on PA performance rather than the negotiation protocols themselves. Additionally, we can also observe that PADOME-based PA achieves up to 34.76% higher average user consent than the alternatives, *e.g.*, Alanezi-based PA in Hospital under BLE. While the effective network communication ranges and space sizes, as noted in [15], naturally influence these results, they can be attributed to the addition of user preference and opponent model elicitation strategies. These strategies, however, result in higher power consumption.

## 5 Discussions and Future Work

The individual strategies employed in PADOME have been previously studied but often under idealized assumptions. In contrast, our work relaxes the assumption of guaranteed user or surrounding PA responses to elicitation requests



and integrates these elicitation strategies, resulting in a more realistic PA design than those proposed previously. A potential direction for future research includes further exploration of novel combinations and adaptive mechanisms building on this integration. This future work should also consider PADOME’s reliance on opportunistic interaction with surrounding PAs, which may not always be feasible in environments with sparse device density or unstable network connectivity. In such cases, reduced access to external models could limit performance.

PADOME achieves significantly higher user consent rates through elicitation of user preferences and opponent models compared to existing PA designs, albeit at the cost of increased power consumption. The increased power consumption raises important considerations for the scalability and applicability of PADOME in large-scale IoT networks, where device battery life and power efficiency could become critical constraints. Future research will need to explore optimization strategies to balance the trade-offs between increased consent rates and power consumption to ensure practical deployment at scale. For example, leveraging edge computing to offload inter-PA and negotiation communication from user smartphones and IoT devices can reduce associated power consumption.

Additionally, PADOME’s adaptiveness is based on predefined heuristics, static probabilistic models, and entropy calculations rather than machine learning (ML) techniques. This design choice is motivated primarily by the resource constraints of IoT and smartphone devices, where running complex learning algorithms is often infeasible. Furthermore, interpretability is especially critical in privacy-sensitive contexts because transparent and explainable decision mechanisms foster user trust and enable informed consent, which ML models often lack due to their black-box nature. Nevertheless, exploring hybrid approaches that incorporate lightweight learning while preserving interpretability represents a promising avenue for future work.

We also note that our experiments are based on the GEPARD simulator, which has limited independent validation. To promote transparency and reproducibility, the simulator’s source code and documentation have been made available, providing explanation of parameter validation and the structure and implementation of the DUNE framework within GEPARD [15]. Future efforts will focus on validation through real-world deployments and benchmarking.

With respect to PADOME’s real-world implementation, it is worth noting that it would not expose its underlying algorithms to end-users. Instead, the interface abstracts these details, presenting only context-specific elicitation requests linked to individual privacy policy statements, as demonstrated in [8,7].

## 6 Conclusion

In this work, we introduced PADOME, a PA with a dynamic user privacy preference and opponent modeling strategies for privacy negotiations in IoT. PADOME combines opponent model elicitation from the surrounding PAs with user preference elicitation that accounts for the possibility of user and other PAs being uncooperative. Using simulations, we demonstrated that PADOME improves aver-

age user consent compared to state-of-the-art PA designs at the cost of increased power consumption. This highlights a critical trade-off between negotiation effectiveness and resource usage. Building on these findings, future research will investigate lightweight, ML-based adaptive opponent modeling and elicitation mechanisms, explore optimization strategies for reducing power consumption, and evaluate PADOME in real-world IoT environments.

## References

1. Alanezi, K., Mishra, S.: Incorporating Individual and Group Privacy Preferences in the Internet of Things. *Journal of AIHC* **13**(4) (2022)
2. Baarslag, T.: What to bid and when to stop. Ph.D. thesis, Delft University of Technology (2014)
3. Baarslag, T., Alan, A.T., et al.: An Automated Negotiation Agent for Permission Management. In: *Proc. AAMAS* (May 2017)
4. Baarslag, T., Gerding, E.H.: Optimal incremental preference elicitation during negotiation. In: *Proc. IJCAI* (Jul 2015)
5. Cha, S.C., Chuang, M.S., et al.: A user-friendly privacy framework for users to achieve consents with nearby BLE devices. *IEEE Access* **6** (2018)
6. Chikukwa, G.: A Consent Framework for the Internet of Things in the GDPR Era. Ph.D. thesis, Dakota State University (2021)
7. Cunche, M., Métayer, D.L., et al.: ColoT: A consent and information assistant for the IoT. In: *Proc. ACM WiSec* (Jul 2020)
8. Das, A., Degeling, M., et al.: Personalized privacy assistants for the internet of things: Providing users with notice and choice. *IEEE Pervasive Computing* **17**(3) (2018)
9. Filipczuk, D., Baarslag, T., et al.: Automated privacy negotiations with preference uncertainty. *Autonomous Agents and Multi-Agent Systems* **36**(2) (2022)
10. Kökciyan, N., Yolum, P., et al.: Taking situation-based privacy decisions: Privacy assistants working with humans. In: *Proc. IJCAI* (Jul 2022)
11. Mehrotra, A., Pejovic, V., et al.: My Phone and Me: Understanding People’s Receptivity to Mobile Notifications. In: *Proc. CHI* (May 2016)
12. Mohammad, Y., Nakadai, S.: Utility elicitation during negotiation with practical elicitation strategies. In: *Proc. IEEE SMC* (Oct 2018)
13. Morel, V., Cunche, M., et al.: A Generic Information and Consent Framework for the IoT. In: *Proc. TrustCom/BigDataSE* (Aug 2019)
14. Padyab, A., Habibipour, A., et al.: Adoption barriers of IoT in large scale pilots. *Information* **11**(1) (2019)
15. Rochester, E., Barker, K.: Designing infrastructure-aware privacy assistants for the iot. Technical report, University of Calgary (2025), <https://hdl.handle.net/1880/122427>, accessed: 2025-08-12
16. Stöver, A., Hahn, S., et al.: Investigating how users imagine their personal privacy assistant. *Proc. PETS* **2** (Jul 2023)
17. Tian, Y., Zhou, K., et al.: What and How long: Prediction of Mobile App Engagement. *ACM Transactions on Information Systems* **40**(1), 1–38 (2022)
18. Van Der Schyff, K., Foster, G., et al.: Online privacy fatigue: a scoping review and research agenda. *Future Internet* **15**(5) (2023)
19. Zhou, H., Goel, M., et al.: Bring Privacy To The Table: Interactive Negotiation for Privacy Settings of Shared Sensing Devices. In: *Proc. SIGCHI* (May 2024)