On the Security of Mutual Authentication Protocols for RFID Systems : The Case of Wei *et al.*'s Protocol

Somitra Kumar Sanadhya
(Joint work with Masoumeh Safkhani, Nasour Bagheri, Majid Naderi and Hamid Behnam)

Assistant Professor,
IIIT-Delhi.
New Delhi, India
somitra@iiitd.ac.in

DPM 2011, KU, Leuven, Belgium
Sep. 15, 2011.

OUTLINE

- General Context.
- Wei *et al.*'s mutual authentication protocol
- Tag impersonation attack
- Desynchronization attacks
- Reader impersonation attack
- Traceability attack
- Strengthening the Wei *et al.*'s protocol
- Closing Remarks

RFID technology

- RFID System = {Tag, Reader, Backend Database}.
- Tags have low cost, low processing power.
- Reader may be capable of heavy computations (It can also access backend DB).
- Widely deployed technology.
  1. Objects in shopping malls,
  2. e-Passports,
  3. Cattle in dairy farms,
  4. Library access control,
  5. Metro travel,
  6. Toll payment on highways and parking,
  7. ....

Reasons for wide deployment of RFID technology

- Provides for identification and authentication of tagged objects.
- Alows data storage and data processing on the tags.
- Data stored on the tags is larger than what other competing methods such as Bar-coding can provide.
- Distance of the RFID reader from the tags can vary from few centimeters to more than 20 meters.
- RFID reader need not be in the line of sight of the tag.

Concerns with RFID systems and solution

1. Concerns: **security and privacy of tag holder**
2. Solving the issue: **mutual authentication protocols:**
   - Authentication is a process in which one party is assured of the identity of another party by obtaining corroborative evidence
   - In our case these parities are the Tag (T) and Reader/back-end database (R).
   - In mutual authentication both the tag and the reader are authenticated to each other

Previous works

- Standardaization: In 2004, the Electronic Product Code Class-1 Generation-2 specification (EPC-C1 G2 in short) was announced by EPC Global which was approved as ISO 18000-6C in July 2006
- The later security analysis has demonstrated important security flaws in the EPC-C1 G2 standard
- Several mutual authentication protocols have been proposed in literature: EPC-friendly protocols, SASI, Gossamer, HB-family,. . .
- Most of them were discovered to be vulnerable to various attacks such as:
  - Secret discloser attack
  - Desynchronization attack
  - Tag impersonation attack
  - Reader impersonation attack
  - Traceability attack

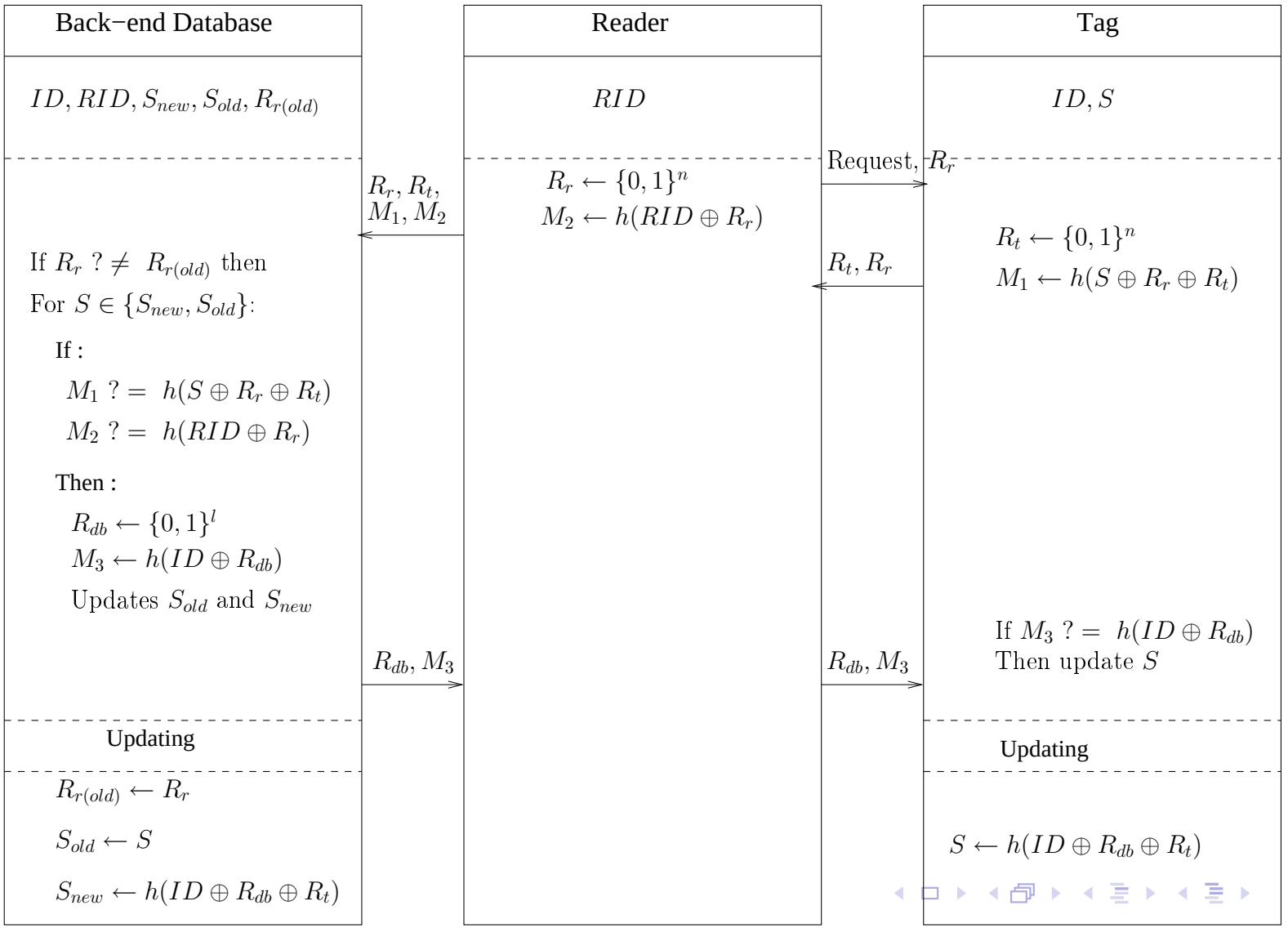
Wei *et al.*'s Protocol
Notation

- R: RFID reader
- T_i: RFID tag i
- B: Back-end database
- $\mathcal{A}$: Adversary
- ID : Unique identifier of tag
- RID : Unique identifier of Reader
- S : A secret value that the database B and the tag T share
- S_{old} : Old value of S
- S_{new} : New value of S
- $R_{r_{(old)}}$: Old reader's random number
- R_t : Tag's random number
- R_{db} : Back-end database's random number
- $h(.)$: One way hash function
- n: The output length of $h(.)$, also equal to the length of secret parameters, e.g. S
- $B \leftarrow A$: Assigning the value of A to B

Wei *et al.*'s Protocol
General points

- The protocol is a hash based mutual authentication protocol which randomizes each authentication session by employing two random values R_r and R_t
- Although most of the other low cost RFID authentication protocols consider the channel between the reader and back-end database secure, the authors consider this channel to be insecure, but:
  - This assumption does not affect our attacks.
  - All the attacks described in the current paper work properly even when we consider this channel secure.

Wei *et al.*'s Protocol Description



Tag impersonation attack
Definition

Tag impersonation attack is a forgery attack in which the reader accepts a spoofed tag as a legitimate tag.

Tag impersonation attack
Learning Phase

1. $\mathcal{A}$ eavesdrops one run of protocol between the reader R and T_i
2. Stores the transferred values between R and T_i
3. The stored values include R_r, $M_1 = h(S \oplus R_r \oplus R_t)$ and R_t
4. At the end of this step, the back-end database assigns the S-value which has been used in the calculation of M_1 to S_{old}

Tag impersonation attack
Impersonation Phase

$\mathcal{A}$ waits until the reader initiates a new session of protocol, where:

1. R sends *Request* and R'_r
2. Once $\mathcal{A}$ receipts the message, it will respond with the tuple R'_t and M'_1, where $R'_t = R_t \oplus R_r \oplus R'_r$ and $M'_1 = M_1 = h(S_{old} \oplus R_r \oplus R_t)$.
3. Once the reader receives R'_t and M'_1:
   1. Computes $M'_2 = h(RID \oplus R'_r)$,
   2. Sends M'_1, M'_2, R'_r and R'_t to the back-end database.
4. Once the back-end database receives M'_1, M'_2, R'_r and R'_t:
   1. Verifies whether $R'_r \stackrel{?}{\neq} R_{r_{(old)}}$ and $M'_1 \stackrel{?}{=} h(S_{old} \oplus R'_r \oplus R'_t)$ and $M'_2 \stackrel{?}{=} h(RID \oplus R'_r)$, where definitely $R'_r \neq R_{r_{(old)}}$ and $M'_2 = h(RID \oplus R'_r)$. On the other hand, $h(S_{old} \oplus R'_r \oplus R'_t) = h(S_{old} \oplus R'_r \oplus R'_r \oplus R_r \oplus R_t) = h(S_{old} \oplus R_r \oplus R_t) = M'_1 = M_1$.
   2. Authenticates $\mathcal{A}$ as a legitimate tag.

Tag impersonation attack
Success probability and complexity

1. The success probability of attack is “1".
2. The complexity of the attack is two runs of the protocol.

Desynchronization attacks
Definition

In desynchronization attack, the adversary forces the tag and the reader to update their common values to different values. If the adversary can succeed in forcing the tag and the reader to do so, they will not authenticate each other in further transactions.

Desynchronization attack
The first attack (Learning Phase)

1. $\mathcal{A}$ eavesdrops one run of the protocol between R and T_i and stores R_r, $M_1 = h(S \oplus R_r \oplus R_t)$ and R_t
2. The back-end database has two record of S, the S value which has been used in the calculation of M_1 denoted by S_{old} and $S_{new} = h(ID \oplus R_{db} \oplus R_t)$.
3. The tag has updated its secret value to $S = h(ID \oplus R_{db} \oplus R_t)$.

Desynchronization attack
The first attack (Desynchronization Phase)

$\mathcal{A}$ waits until a new session of protocol to be initiated, where:

1. R sends $Request$ and R'_r.
2. Once $\mathcal{A}$ receipts the message, it will respond with the tuple R'_t and M'_1, where $R'_t = R_t \oplus R_r \oplus R'_r$ and $M'_1 = M_1 = h(S_{old} \oplus R_r \oplus R_t)$.
3. The back-end database authenticates $\mathcal{A}$ as the legitimate tag T_i, generates a new random value R'_{db} and updates its secret value. However, since S_{old} has been used throughout the authentication process, the reader R assigns it to S_{old} and updates $S'_{new} = h(ID \oplus R'_{db} \oplus R'_t) = h(ID \oplus R'_{db} \oplus R'_r \oplus R_r \oplus R_t)$.
4. Since the secret key S in tag remains $h(ID \oplus R_{db} \oplus R_t)$, if $S \neq S_{old}$ and $S \neq S'_{new}$, then R and T_i have different secret values in their database and they will not authenticate each other any more.

Desynchronization attack
Success probability and complexity

1. The probability of $S = S_{old}$ is 2^{-n}.
2. Similarly, the probability for $S = S'_{new}$ is also 2^{-n}.
3. Therefore, The success probability of the given attack is $1 - 2^{-(n-1)}$.
4. The complexity of the attack is two runs of the protocol.

Desynchronization attack
The second attack (Learning Phase)

1. $\mathcal{A}$ eavesdrops one run of protocol between the reader R and T_i
2. $\mathcal{A}$ stores the transferred values between R and T_i at the last step of protocol, $M_3 = h(R_{db} \oplus ID)$ and R_{db}.
3. At the end of this step, the back-end database has two records of S, S_{old} and $S_{new} = h(ID \oplus R_{db} \oplus R_t)$ and the tag updates its secret value to $S = h(ID \oplus R_{db} \oplus R_t)$.

Desynchronization attack
The second attack (Desynchronization Phase)

1. On the next session of protocol, $\mathcal{A}$ does not disturb the first and the second steps of the protocol.
2. On the third step of protocol, the back-end database authenticates T_i, generates a new random number R'_{db}, assigns the current secret $S_{new} = h(ID \oplus R_{db} \oplus R_t)$ to S_{old}, updates $S'_{new} = h(ID \oplus R'_{db} \oplus R'_t)$ and sends $M'_3 = h(R'_{db} \oplus ID)$ and R'_{db} to R to forward them to T_i.
3. $\mathcal{A}$ blocks $M'_3 = h(R'_{db} \oplus ID)$ and R'_{db} and instead sends $M_3 = h(R_{db} \oplus ID)$ and R_{db} to T_i.
4. As the tag receives M_3 and R_{db} it does as follows:
   1. Verifies whether $M_3 \stackrel{?}{=} h(ID \oplus R_{db})$, which it is,
      - Authenticates the reader,
      - Updates $S'_{new} = h(ID \oplus R_{db} \oplus R'_t)$
5. Now, the records of secret value S that back-end database has are $S_{old} = h(ID \oplus R_{db} \oplus R_t)$ and $S'_{new} = h(ID \oplus R'_{db} \oplus R'_t)$, while the secret value stored in T_i is $S = h(ID \oplus R_{db} \oplus R'_t)$.

Desynchronization attack
Success probability and complexity

1. The probability of $S = S_{old}$ is 2^{-n}.
2. Similarly, the probability for $S = S'_{new}$ is also 2^{-n}.
3. Therefore, The success probability of the given attack is $1 - 2^{-(n-1)}$.
4. The complexity of the attack is two runs of the protocol.

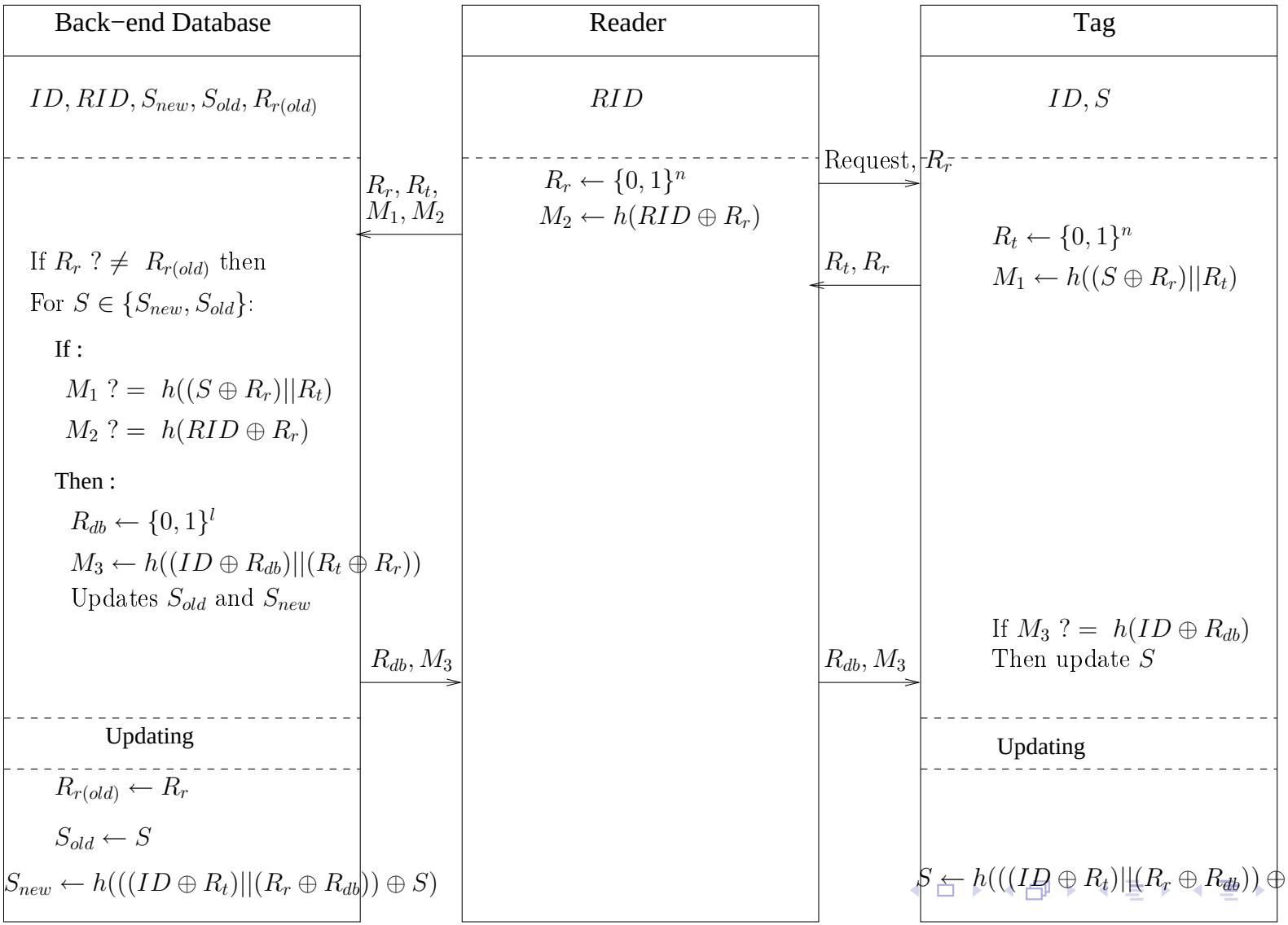
Reader impersonation attack and Traceability attack

- The second desynchronization attack can be used as a reader impersonation attack:
  1. $\mathcal{A}$ eavesdrops tuple M_3 and R_{db} in one session
  2. $\mathcal{A}$ supplants a new session
  3. At the last step of the protocol, $\mathcal{A}$ replies with the eavesdropped M_3 and R_{db}
  4. Definitely, T_i authenticates $\mathcal{A}$ as a legitimate reader
  5. The success probability of the attack is “1” and the complexity is two runs of protocol.

Traceability attack

- The second desynchronization attack can be used as a Traceability attack:
  1. $\mathcal{A}$ eavesdrops tuple M_3 and R_{db} in one session of the protocol that T_i has been involved in
  2. Given a tag T_j, the adversary will try to impersonate the reader based on the eavesdropped M_3 and R_{db}
  3. If T_j authenticated $\mathcal{A}$ then adversary concludes that the given tag is T_i, otherwise it is not
  4. The success probability of the attack is $(1 - 2^{-(n)})$ and complexity is two runs of protocol

Strengthening Wei *et al.*'s Protocol Description



Closing Remarks

1. We present several efficient attacks against Wei *et al.*'s mutual authentication protocol.
2. It is the first attack in the literature against this protocol
3. We proposed strengthened version of Wei *et al.*'s protocol
4. We replaced XOR operation in the protocol by the concatenation which extends the length of the message that must be hashed.
5. The modified protocol will need few extra calls to the underlying compression function which will increase the computational complexity of the protocol.
6. This increase in the complexity is providing more security.

Thank You