Short Papers

Security Analysis of the SASI Protocol

Tianjie Cao, Elisa Bertino, Fellow, IEEE, and Hong Lei

Abstract—The ultralightweight RFID protocols only involve simple bitwise operations (like XOR, AND, OR, etc.) on tags. In this paper, we show that the ultralightweight strong authentication and strong integrity (SASI) protocol has two security vulnerabilities, namely, denial-of-service and anonymity tracing based on a compromised tag. The former permanently disables the authentication capability of a RFID tag by destroying synchronization between the tag and the RFID reader. The latter links a compromised tag with past actions performed on this tag.

Index Terms—Authentication, location-dependent and sensitive, security and privacy protection.

1 INTRODUCTION

Radio Frequency Identification (RFID) systems are a common and useful tool for admission control, payment, ticketing, and supply chain management. However, several security and privacy concerns have been identified in connection with the use of RFIDs.

A RFID system typically consists of two components: a set of tags, also called transponders, and a set of readers, also called transceivers. Tags are attached to physical objects. Readers query these tags for some (potentially unique) identifying information about the objects to which tags are attached. Although readers are often regarded as a simple conduit to a back-end database, for simplicity, we treat a reader and a back-end database as a single entity. A key security problem in such a context is that an adversary can arbitrarily modify the conversations between any pair of tag and reader and indeed initiates and terminates a session at its choice.

The security of an RFID protocol can be described in terms of four games, an authentication game $G_{auth}$, an anonymity game $G_{anon}$, a forward anonymity game $G_{fanon}$ and an availability game $G_{avail}$, with the following players: the malicious adversary $A$ against the honest tags and the honest readers. These games have two steps. The first step is a preparing step for adversary $A$: $A$ is allowed to interact arbitrarily with the tags and the readers. In the second step, $A$’s knowledge is tested. The score of $A$ in game $G$ is its advantage $\text{adv}^{G}_{A}$. $A$ wins if its advantage is non-negligible. We now describe in more detail the second steps of the four games: $G_{auth}$, $G_{anon}$, $G_{fanon}$, and $G_{avail}$.

Authentication. In the second step of $G_{auth}$, $A$ must impersonate some tag $T$ to some reader $R$. During this impersonation step, $A$ is allowed to interact arbitrarily with all other tags and readers, except tag $T$ that $A$ is trying to impersonate. The advantage $\text{adv}^{G_{auth}}_{A}$ of the adversary is the probability that $A$ succeeds in authenticating itself to $R$. An RFID protocol is a secure authentication protocol if $\text{adv}^{G_{auth}}_{A}$ is negligible. Impersonation is an attack on authentication.

Anonymity. Anonymity means that given two interactions $A$ is not able to say whether they are with the same tag $T$. For anonymity, we require that the advantage $\text{adv}^{G_{anon}}_{A}$ of the adversary in the second step of $G_{anon}$ in linking different interactions with the same tag is negligible. Anonymity property is also called untraceability, unlinkability, or indistinguishability.

Forward anonymity. Forward anonymity means that even if the adversary obtains the secret data stored at a tag by tampering with the tag, the adversary’s advantage $\text{adv}^{G_{fanon}}_{A}$ in the second step of $G_{fanon}$ in tracing the data back is negligible. The authentication transcripts of the tag should not be traced back using previous known messages, i.e., disclosed data and communication information. Forward anonymity is often called forward security or forward untraceability. Tracing is an attack on forward anonymity.

Availability. In $G_{avail}$, the adversary $A$ must prevent a tag $T$ from being authenticated by a reader $R$ in a challenge session $ses$, without interacting with the session $ses$. In this attack, $A$ is allowed to interact with all tags and readers, except of course for the session $ses$. The advantage $\text{adv}^{G_{avail}}_{A}$ of $A$ in this game is the probability that $R$ rejects $T$ in the challenge session $ses$. An RFID protocol is an availability-assuring protocol if $\text{adv}^{G_{avail}}_{A}$ is negligible. Denial of service (DoS) is an attack against the availability. Especially, the desynchronization attack carried out by a man-in-the-middle attack must be prevented.

To deal with the above security threats, many authentication protocols for RFID tags have been proposed so far. RFID tags are generally low cost with extremely limited resources, so they cannot perform the public key algorithms. Most previous protocols require the support of either hash function or symmetric encryption on the tag. The lightweight RFID authentication protocols require a random number generator and simple functions like cyclic redundancy code (CRC) checksum [1], [2], [3], [4], [5]. Some weaknesses of these schemes have been recently reported [6], [7], [8]. The ultralightweight protocols only involve simple bitwise operations (like XOR, AND, OR, etc.) on tags [9], [10], [11], [12], [13], [14]. However, desynchronization attack and the full-disclosure attack against such protocols have been reported [9], [10], [11].

Recently, Chien proposed the ultralightweight strong authentication and strong integrity (SASI) protocol, where the tag requires only simple bitwise operations [15]. Since the tag does not support random number generator to generate a challenge nonce, an attacker can replay old messages and impersonates a reader. Thus, the assertion of the SASI protocol that it provides mutual authentication is incorrect. In [15], Chien claimed that the SASI protocol is resistant to the desynchronization attack and man-in-the-middle attack. However, we show that the SASI protocol is prone to DoS attacks. In our attacks, a man-in-the-middle can destroy the synchronization between the database and the tag. Thus, the tag cannot be further authenticated by the database. The RFID system will be involved in DoS state and unable to guarantee availability. Chien also claimed that the SASI protocol satisfies forward security. However, if we assume that an attacker compromises a tag, the attacker can infer the previous secret data and keys of the same tag and trace the past communication. Thus, the SASI protocol does not provide forward anonymity.

The rest of this paper is organized as follows: We review the SASI protocol in Section 2 and analyze two vulnerabilities in Section 3. In Section 4, we conclude the paper.
2 Review of the SASI Protocol

In the SASI protocol [15], each tag has a static identifier (ID) and preshares a pseudonym (IDS) and two keys $K_1, K_2$ with the back-end database. The length of each of ID, IDS, $K_1$, and $K_2$ is $n$ bits. Typically, the value $n$ is 96. Each tag keeps two entries of the form (IDS, $K_1, K_2$): one is for the old values for the pseudonym and two keys, and the other is for the potential next values. SASI is a highly efficient RFID authentication protocol using only bitwise XOR ($\oplus$), bitwise OR ($\lor$), bitwise AND ($\land$), addition mod $2^n$ ($+$), and left rotate ($\text{Rot}(x,y)$) operations. $\text{Rot}(x,y)$ left rotates the value of $x$ by $y$ bits. Expensive operations, such as hash functions, are not required at all by SASI, and random number generation is only executed by the reader. SASI assumes that the channel between the reader and the back-end database is secure, but the channel between the reader and the tag is susceptible to all the possible attacks. The specification of the SASI protocol is shown in Fig. 1.

![Fig. 1. The SASI protocol.](image)

The protocol has three phases: tag identification, mutual authentication, and pseudonym update and key update.

3 Vulnerabilities of the SASI Protocol

We assume that there is a completion message exchanged between the tag and the reader to indicate a successful completion of the protocol. This completion message will enable the update operations at both the reader and the tag.

3.1 DoS Attack

In general, a DoS attack results in loss of service to users. In other words, the attacker does not try to obtain information, but rather, it tries to prevent a legitimate reader from accessing data stored in tags. To assure untraceability for a RFID tag, the SASI protocol updates the database’s secret information, that is, IDS, $K_1$, and $K_2$ after a successful protocol run. The tag updates the secret information accordingly so that a reader can still authenticate the tag later on. Therefore, the synchronization of secret information between the database and the tag is crucial to resist to DoS attacks.

In the SASI protocol, if the current secret information $K_{1_{\text{next}}}$, $K_{2_{\text{next}}}$, $K_{1_{\text{old}}}$, and $K_{2_{\text{old}}}$ for a tag is different from the key $K_1$ and $K_2$ stored in the database, the tag will be in a desynchronization state with respect to the database leading to a DoS situation. In what follows, we show attacks that lead to a desynchronization state for the tag.

**Attack 1: Changing messages A, C, and D.** An attacker can first eavesdrop on the on-going protocol and then replace $A \parallel B \parallel C$ with $A' \parallel B' \parallel C'$, where $A' = A \oplus [I]_0$, $C' = C \oplus [I]_0$, and $[I]_0 = [000...001]$ (set the first $n - 1$ most significant bits of $I$ as 0 and the least significant bit as 1). Similarly, the attacker changes the reply $D$ from the tag to $D' = D \oplus [I]_0$. This procedure is specified in Table 1.

### Table 1: Changing Messages A, C, and D

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>Reader→Tag: hello</td>
</tr>
<tr>
<td>$B$</td>
<td>Tag→Reader: IDS</td>
</tr>
<tr>
<td>$C$</td>
<td>Reader→Tag(Attacker): $A' \parallel B' \parallel C'$</td>
</tr>
<tr>
<td>$D$</td>
<td>Tag(Attacker)→Tag: $D'$</td>
</tr>
</tbody>
</table>

**Tag identification.** Initially, the reader sends “hello” to the tag, which then responds with its potential next IDS. If the reader can find a matched entry in the database, it starts the mutual authentication phase; otherwise, it probes the tag again, and the tag responds with its old IDS.

**Mutual authentication.** The reader uses the matched values and two randomly generated integers $n_1$ and $n_2$ to compute the values $A$, $B$, and $C$ (the calculation equations are specified in Fig. 1). Such values are then sent to the tag. From $A \parallel B \parallel C$, the tag first extracts $n_1$ from $A$, extracts $n_2$ from $B$, computes $\overline{K}_1$, $\overline{K}_2$, and then computes the response value $D$. Upon receiving $D$, the reader uses its local values to verify $D$.

**Pseudonym update and key update.** After the reader and the tag authenticate each other, they update their local pseudonym and keys.
We now analyze the success rate of such an attack:

1. Once the tag receives $A'||B'||C'$, the probability that the tag accepts the message $A'||B'||C'$ is not less than $1/(2n)$.

   Suppose that $K2$ is a random number; there is a probability equal to $1/n$ that $K2 \mod n = 0$ and a $1/2$ probability that the least significant bit of $K2 = \overline{K1}$ is 0. We note that for any $X$ there is $\text{Rot}(K2 \oplus X, K2) = K2 \oplus X$ when $K2 \mod n = 0$. In this case, we check the validity of the message $A'||B'||C'$:

   \[
   C' = C \oplus |I|_0
   = [(K1 \oplus \overline{K2}) + (\overline{K1} \oplus K2)] \oplus |I|_0
   = (K1 \oplus \overline{K2} \oplus |I|_0) \oplus (\overline{K1} \oplus K2).
   \]

   $n1' = A' \oplus IDS \oplus K1$
   $= A \oplus |I|_0 \oplus IDS \oplus K1$
   $= (IDS \oplus K1 \oplus n1) \oplus |I|_0 \oplus IDS \oplus K1$
   $= n1 \oplus |I|_0$.

   The operation on $A$ is actually toggling the least significant bit of $n1$:

   \[
   n2' = B - (IDS \lor K2)
   = (IDS \lor K2) + n2 - (IDS \lor K2)
   = n2.
   \]

   $\overline{K1}' = \text{Rot}(K1 \oplus n2, K1) = \overline{K1}$,
   $\overline{K2}' = \text{Rot}(K2 \oplus n1', K2) = K2 \oplus n1' = \overline{K2} \oplus |I|_0$.
   \[
   \tilde{C} = (K1 \oplus \overline{K2}') + (\overline{K1}' \oplus K2)
   = (K1 \oplus \overline{K2} \oplus |I|_0) \oplus (\overline{K1} \oplus K2)
   = C'.
   \]

   In the case in which $K2 \mod n = 0$ and the least significant bit of $K2 \oplus \overline{K1}$ is 0, the tag will accept the message $A'||B'||C'$.

2. Once the reader receives $D'$, the probability that the reader accepts the message $D'$ is not less than $1/2$.

   If the least significant bit of $ID$ is 0, the reader will accept $D'$. There is a $1/2$ probability that the least significant bit of $ID$ is 0. We have

   \[
   D' = D \oplus |I|_0
   = (\overline{K2}' + ID) \oplus ((K1 \oplus K2) \lor \overline{K1}) \oplus |I|_0
   = (\overline{K2} \oplus |I|_0 + ID) \oplus ((K1 \oplus K2) \lor \overline{K1}) \oplus |I|_0
   = (\overline{K2} \oplus |I|_0 + ID) \oplus (K1 \oplus K2 \lor \overline{K1})
   = (\overline{K2} + ID) \oplus ((K1 \oplus K2) \lor \overline{K1})
   = \tilde{D}.
   \]

   Once the reader accepts the value, the reader needs to update the tag's secret information with the pair $(n1, n2)$. However, the tag uses another pair $(n1 \oplus |I|_0, n2)$ to update its secrets. It is obvious that there is a mismatch between the secrets stored at the tag and at the reader. Therefore, there is a non-negligible probability value, that is, $(1/n) \times (1/2) \times (1/2) = 1/(4n)$ succeeding in a DoS attack. In fact, this attack can be extended to toggle a single bit of $A$ at any location $i$, so that it can be a general attack with the same $1/(4n)$ success probability.

**Attack 2: Changing messages $B$ and $C$.** The attacker can first eavesdrop on the on-going protocol and then replace $A||B||C$ with $A||B'||C'$, where $B' = B + 1$, $C' = C \oplus |I|_0$. This procedure is specified in Table 2.

At the tag side, the attack does not affect the first round of the interaction protocol, that is, “tag identification.” In the second round, when the tag receives the message $A||B'||C'$, it can still authenticate the reader with a non-negligible probability. But, the tag will receive a wrong random number $n2'$ (where $n2'$ depends on $n2$). The tag will accept this value and compute its reply according to $n2'$. In this attack, the attacker can now provide the reader with a reply $D$. If the reader accepts value $D$, the attack is successful; otherwise, the attack fails. Now, we analyze the success rate:

1. Once the tag receives $A||B'||C'$, the probability that the tag accepts the message $A||B'||C'$ is not less than $1/(4n)$.

   Suppose that $K1$ is a random number, there is a probability equal to $1/n$ that $K1 \mod n = 0$ and a $1/4$ probability that the least significant bit of $K1 \oplus \overline{K2}$ and $n2$ are 0 simultaneously. We note that for any $X$ there is $\text{Rot}(K1 \oplus X, K1) = K1 \oplus X$ when $K1 \mod n = 0$. In this case, we check the validity of the message $A||B'||C'$:

   \[
   C' = C \oplus |I|_0
   = [(K1 \oplus \overline{K2}) + (\overline{K1} \oplus K2)] \oplus |I|_0
   = (K1 \oplus \overline{K2} \oplus |I|_0) \oplus (\overline{K1} \oplus K2).
   \]

   $n1' = A' \oplus IDS \oplus K1$
   $= (IDS \oplus K1 \oplus n1) \oplus IDS \oplus K1$
   $= n1$,

   $n2' = B - (IDS \lor K2)$
   $= B + 1 - (IDS \lor K2)$
   $= ((IDS \lor K2) + n2) + 1 - (IDS \lor K2)$
   $= ((IDS \lor K2) + (n2 \oplus |I|_0)) - (IDS \lor K2)$
   $= n2 \oplus |I|_0$.

   The operation on $B$ is actually toggling the least significant bit of $n2$:

   \[
   \overline{K1}' = \text{Rot}(K1 \oplus n2', K1) = K1 \oplus n2' = \overline{K1} \oplus |I|_0,
   \overline{K2}' = \text{Rot}(K2 \oplus n1', K2) = \overline{K2},
   \tilde{C} = (K1 \oplus \overline{K2}') + (\overline{K1}' \oplus K2)
   = (K1 \oplus \overline{K2}) + (\overline{K1} \oplus |I|_0 \oplus K2)
   = C'.
   \]

---

### Table 2

<table>
<thead>
<tr>
<th>Reader→Tag: hello</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag→Reader: IDS</td>
</tr>
<tr>
<td>Reader→Tag(Attacker):</td>
</tr>
<tr>
<td>$A</td>
</tr>
<tr>
<td>Reader(Attacker)→Tag:</td>
</tr>
<tr>
<td>$A</td>
</tr>
<tr>
<td>Tag→Reader: D</td>
</tr>
</tbody>
</table>

**where:**

- $B' = B + 1$
- $C' = C \oplus |I|_0$
### Table 3
Changing $A$ and Guessing $C$

<table>
<thead>
<tr>
<th>$i$</th>
<th>$A$ and $C$ values</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$A'_{0}$</td>
<td>$C'_{0}$</td>
</tr>
<tr>
<td>$j$</td>
<td>$A'_{j}$</td>
<td>$C'_{j}$</td>
</tr>
</tbody>
</table>

\[ D = (\overline{K}'_{2} + ID) \oplus ((K1 \oplus K2) \lor \overline{K}') \]
\[ = (\overline{K}_{2} + ID) \oplus ((K1 \oplus K2) \lor \overline{K}_{1}) \]
\[ = \overline{D}. \]

### Table 4
Tracing Attack

| $i$ | Trace of $A' || B || C_{0}$ from view database |
|-----|---------------------------------------------|
| 0   | $(IDS_{0}, A_{0}, B_{0}, C_{0}, D_{0})$ |

\[
\text{LET } i = 2 \mod n, \text{ then } \overline{C}_{0} = C + [I]_{i}, \text{ where } C_{0} = C - [I]_{i}.
\]

Let $i = K2 \mod n$, we prove that the tag will accept $A' || B || C_{0}$ or $A' || B || C_{1}$, where $A' = A \oplus [I]_{i}, C_{0} = C + [I]_{i}$, and $C_{1} = C - [I]_{i}$. Similar to the analysis in attack 1, we have

\[ n1' = n1 \oplus [I]_{i}, \]
\[ n2' = n2, \]
\[ \overline{K}' = \text{Rot}(K1 \oplus n2, K1) = \overline{K}_{1}, \]
\[ \overline{K}_{2}' = \text{Rot}(K2 \oplus n1', K2) \]
\[ = \text{Rot}(K2 \oplus n1 \oplus [I]_{i}, K2) \]
\[ = \text{Rot}(K2 \oplus n1 \oplus [I]_{i}, K2) \]
\[ = \text{Rot}(K2 \oplus n1 \oplus [I]_{i}, K2) \]
\[ = \overline{K}_{2}. \]
\[ C_{0} = C + [I]_{i}, \]
\[ C_{1} = C - [I]_{i}. \]

If the $i$th least significant bit of $K1 \oplus K2$ is 0, then $\overline{C}$ is equal to $C_{0}$, else $\overline{C}$ is equal to $C_{1}$. Once the tag accepts the guessed value, the tag will update the tag’s secret information with the pair $(n1 \oplus [I]_{i}, n2)$. However, the reader has updated the secrets with another pair $(n1, n2)$. It is obvious that there is a mismatch between the secrets stored at the tag and at the reader. There are at most $2n$ guesses to succeed in such DoS attack.

#### 3.2 Tracing Attack
RFID tags are inexpensive devices that offer no tamper resistance. Hence, an attacker upon compromising a tag may be able to read its secret values and link this tag with past actions performed on the tag. With forward anonymity, disclosure of current secret key material does not compromise the secrecy of earlier material. The SASI protocol does not provide forward anonymity. The details of the attack are given in Table 4.

A communication view of the protocol is defined to be the set of all messages that the reader has received and generated when authenticating a tag. The attacker can construct the communication view database by recording all the authentication transcripts between the reader and the tags. For each instance $i$ of the protocol,
the attacker can record \((IDS_1, A, B, C, D_1)\) when the tag communicates with the reader during the instance \(i\) of the protocol. The tuple \((IDS_1, A, B, C, D_1)\) is referred to as the view by the attacker on the instance \(i\) of the protocol. We suppose that the communication view database has \(N\) records.

Suppose that a tag is compromised through a physical attack after the authentication phase. Then, the attacker would get the values \(ID, IDS_m, K_{1m}, K_{2m}, IDS_{m+1}, K_{1m+1}, K_{2m+1}\) of the tag. To link the values \((ID, IDS_{m+1}, K_{1m+1}, K_{2m+1})\) and the past communication, the attacker can easily trace the last authentication view by searching the communication view database using the condition \(IDS = IDS_m\). Now, we introduce an algorithm to find the tuple record in the communication view database that links to \((ID, IDS_m, K_{1m}, K_{2m})\).

For each view record \((IDS_1, A, B, C, D_1)\), the attacker checks whether or not it links to \((ID, IDS_m, K_{1m}, K_{2m})\). The attacker computes \((IDS_1 \oplus (IDS_1 + ID)) \oplus K_{1m}\) to derive the value \(n2\). Once the attack obtains \(n2\), it can compute \(K_{1m}\) from equation \(K_{1m} = \text{Rot}(K_{1m-1} \oplus n2, K_{1m-1})\). The attacker computes all potential candidates \(K_{1m-1} = \text{Rot}(K_{1m-1} \oplus j, K_{1m-1})\) for \(0 \leq j \leq n - 1\) and checks whether \(K_{1m}\) is equal to \(\text{Rot}(K_{1m-1} \oplus n2, K_{1m-1})\). If such two values match, the attacker computes \(K_{2m-1} = (C_i - (K_{1m-1} \oplus K_{2m})) \oplus K_{1m}\) and \(n1 = A_i \oplus IDS_1 \oplus K_{1m-1}\) and checks whether \(B_i\) is equal to \((IDS_1 \oplus V_{K_{2m-1}} + n2)\) and \(D_i\) is equal to \((K_{2m} + ID) \oplus ((K_{1m-1} \oplus K_{2m-1}) \oplus K_{1m})\). If the above equations hold, the attacker succeeds. Otherwise, the attacker checks the next view record. Once the attacker obtains \((i, IDS_i, K_{1m-1}, K_{2m-1})\), it can use the same algorithm to trace the former communication view. Therefore, the SASI protocol does not provide forward anonymity; the past communication from the same tag can be traced.

4 CONCLUSIONS

In this paper, we have demonstrated two effective attacks against the SASI protocol recently proposed in [15]. The severity of the attacks indicates the insecure design of the protocol. Our work shows that it may be quite dangerous using only simple bitwise operations to achieve RFID authentication under powerful adversarial model. The security of such protocols must be proved with careful cryptanalysis. How to design a secure protocol without strong cryptographic algorithms such as hash function and symmetric encryption is an open problem. We plan, as our next step, to design a secure (ultra) lightweight RFID mutual authentication protocol that keeps these attacks into account and to apply it to low-cost RFID tags.

ACKNOWLEDGMENTS

The authors would like to thank the anonymous reviewers for their helpful comments. This work is partially supported by the US National Science Foundation under Grant 0712846 “IPS: Security Services for Healthcare Applications,” the Jiangsu Provincial Natural Science Foundation of China (BK2007035), and the Science and Technology Foundation of CUMT.

REFERENCES