International Conference on Information Security & Privacy (ICISP2015), 11-12 December 2015, Nagpur, INDIA

A secure authentication scheme for RFID systems

Prajnamaya Dassa\textsuperscript{a,}\textsuperscript{*}, Hari Om\textsuperscript{b}

\textsuperscript{a,b}Department of Computer Science and Engineering, Indian School Mines, Dhanbad, Jharkhand, 826004

Abstract

Day by day the importance of Radio Frequency Identification (RFID) systems is increasing for its powerful capabilities in automatic identification, localization and access control of the objects. However, the RFID techniques are plagued to security and privacy issues due to underlying wireless communication channel. In order to come up with a solution, we propose an efficient authentication scheme which uses pseudorandom number generators (PRNG) and some simple cryptographic operations. Moreover, as the current generation tags come with in-built pseudo random generators, the implementations of these operations are possible with low complexity. The secret information stored inside the tags is communicated in a more secure way ensuring confidentiality, integrity, and authentication. The security of our proposed scheme is analyzed against different attacks on RFID and with the performance of some existing protocols. Experimental results show a significant improvement in security with average cost, when compared with the existing techniques.

© 2016 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/).

Peer-review under responsibility of organizing committee of the ICISP2015

Keywords: Secret; Authentication; Pseudorandom number generator; RFID; Attack.

1. Introduction

Technological advancements in the field of wireless communication have challenged most of the communications to operate without human intervention. RFID is a fast pervading wireless data collection technology having radio transmission that contains some identifying information about the objects for automatic identification. In RFID systems, the devices use electromagnetic fields wirelessly to exchange the identifying data. This technology is able

\textsuperscript{*} Corresponding author. Tel.: +91-9437646965 / 8804754214

E-mail address: prajnamayadass@gmail.com
to overcome the problem of line-of-sight communication involved in bar code technology and now an emerging type of network that plays an important role in the Internet-of-Things (IoT)\(^1\). An RFID system comprises of tag, reader, and backend server as its main components. Tags, the basic building block of RFID, consist of a chip, antenna and a certain amount of computational and storage capabilities. A reader queries tag to obtain tag contents and sends the encrypted information received from the tag to the backend server for checking the legitimacy of the tag. The backend server contains a local database and some processors. The tags, the transponders, are queried by readers through a wireless insecure channel. The underlying channel between a reader and the backend server may be wired or wireless and is secured\(^1\) as shown in Fig. 1. The tags may be classified as passive, semi passive, and active, according to the way they are powered. In this paper, we have considered passive tags that require no internal power but are powered from the radio signal sent from the reader while they query the tag\(^2\).

As the passive tags have low storage and low computational capabilities, they suffer from many security flaws. Due to the limitations on storage, RFID authentication protocols use low cost cryptographic primitives like bitwise operations, pseudorandom number generators, hash functions etc\(^3, 4\)\(...\). In the proposed scheme, pseudo random number generators are mainly used to provide secure authentication. The requirement of pseudo-randomness is based on the fact that the output of a random generator should be indistinguishable from uniformly distributed random variables. Randomness in seed value also plays an important role and it is very hard to find pseudorandom stream if the seed value is not known\(^6\). In our scheme, we use random numbers along with the secret value of tag as seed and update the secret of the tag after each authentication. The randomness property of the generators helps providing confidentiality and immunity to replay attack\(^8\). As the secret is updated after each session, forward secrecy is also ensured.

The rest of the paper is structured as follows. In Section 2, the reviews of related work have been discussed. Section 3, describes the proposed authentication scheme, using pseudo-randomness property. Section 4 presents security and privacy analysis. In section 5, the efficiency of the proposed scheme has been calculated and compared against some existing schemes. Finally, we conclude our paper in section 6.

2. Related Work

Several authentication schemes have been proposed for providing security in RFID systems. Here, we give a brief review of some protocols. Albert et al.\(^2\) use three phases for authentication: initialization, identification (synchronised and desynchronised), and updating phase. This protocol uses the PRNGs to produce an unpredictable pseudo-random sequence. In Fu et al.\(^5\), a scalable pseudo random based mutual authentication scheme is proposed that involves encryption based on symmetric key cryptography, random number generators, and hash function. It uses an alias of tag ID, updates it in every authentication, and requires EEPROM memory for this purpose. Min et al.\(^7\) discuss a dynamic token based authentication scheme. This protocol provides anonymity and authentication through random initialization of tokens and these tokens are updated dynamically through the use of the base token.
and base indicator array used. This scheme however requires more storage in the tag for these arrays and updates. It does not support mutual authentication and prone to desynchronization attack. In the scheme of Gui et al.9, a hash function based key encryption is used for authentication and ownership transfer. This encryption function divides the tags into several groups and sequence of the tag requires to be checked. Irfan et al.10 proposed an authentication scheme based on cryptographic hash chain. But using the hash recursively more times requires more storage and computations. The protocol proposed by Chang et al.11 is an improved version to the protocol by Venkatraman et al.14. They have detected some privacy concerns in the Venkatraman et al.’s protocol14 and proposed a new scheme without the RFID middleware. In the Chang et al.’s scheme11, an authentication process is carried out by considering the previous session termination. Rahman et al.12 discussed a protocol based on PRNG but it assumes that reader has all the secrets before authentication process starts and limits its applications. Li et al.13 provide an authentication scheme that uses pseudo-random generators at reader side and performs ID and key updating on every successful authentication. However, this scheme is not secure against the traceability attack.

3. Proposed authentication scheme

Here we propose our scheme for securing an RFID system. It uses light cryptographic operations like PRNGs, hash functions, and XORs. We mention the assumptions and notations used in our scheme, followed by the authentication process.

3.1 Assumptions

We assume the RFID system with three entities: Tag, RFID reader, and backend server.

- The communication channel between reader and backend server is fully secure. So we do not consider any security between them.
- The tag is a passive device and communicates with reader through an insecure channel.
- The tag contains two data fields (S, ID) i.e., secret of tag and tag pseudonym (index value in database). S is of 128 bits and ID is 96 bits in length.
- As S is updated in tag, we require rewritable memory (EEPROM or FRAM) of 128 bits.
- The backend server contains a local database that contains fields: ID, h(ID), S_{Old}, S_{New}. Initially, the S_{Old} contains zero and S_{New} contains the secret value of tag.
- Random numbers generated are of 96 bits in length.
- Tag memory is insecure and vulnerable to physical attacks.

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>S, ID</td>
<td>Secret and ID pseudonym of Tag</td>
</tr>
<tr>
<td>S_{Old}, S_{New}</td>
<td>Current and previous session secrets of tag stored in backend server</td>
</tr>
<tr>
<td>h()</td>
<td>One-way hash function</td>
</tr>
<tr>
<td>PRNG(A)</td>
<td>PRNG to find a random number with A as seed value</td>
</tr>
<tr>
<td>PRNG(A,B)</td>
<td>PRNG to find B^A Pseudo-random number with A as seed value</td>
</tr>
<tr>
<td>( | )</td>
<td>Function of string concatenation</td>
</tr>
<tr>
<td>( \oplus )</td>
<td>XOR operation</td>
</tr>
</tbody>
</table>

3.2 Authentication process

The reader initiates the authentication process by querying the tag. The reader first selects a tag and then requests its information. The whole process is described below and depicted in Fig. 2.
(1) Reader generates a random number ($N_R$) and sends it to RFID tag.

(2) After Receiving $N_R$ from reader, the tag generates another random number ($N_T$). The tag then calculates $V = PRNG(S \oplus N_R \oplus N_T)$ and $H = h(\text{ID})$, using the information (Tag ID, $S$) stored in tag. The tag sends $V$, $H$, and $N_T$ to reader. The reader forwards $V$, $H$, and $N_T$ received from tag to server along with $N_R$ generated by it.

(3) The server after receiving these values, retrieves database records of $h(\text{ID})$ to find if there is a record corresponding to $H$.
   - If no record is there corresponding to that value, then communication is terminated.
   - If a record is found, the server extracts the corresponding tag’s secret $S_{\text{New}}$ from database and calculates $V' = PRNG(S_{\text{New}} \oplus N_R \oplus N_T)$ to verify if $V'$ and the received $V$ are identical or not. If they are equal, it is confirmed that the previous session was successful and tag contains $S_{\text{New}}$ as its $S$ value. The server sends $S_{\text{New}}$ to reader.
   - If $V'$ and $V$ are not equal, server extracts $S_{\text{Old}}$ value for the corresponding matched tag and calculates $V'' = PRNG(S_{\text{Old}} \oplus N_R \oplus N_T)$. It checks if $V''$ and $V$ are equal. If equality holds, then it makes the variable ‘Flag’=1 and sends $S_{\text{Old}}$ to reader. Here, the server is confirmed about the unsuccessful previous session and the tag contains $S_{\text{Old}}$ as its $S$ value.

(4) The reader takes the value (either $S_{\text{New}}$ or $S_{\text{Old}}$) as the seed from server and calculates $M = N_R^{th}$ pseudorandom number. Using $M$ as seed value, it calculates a random number ($N$) from PRNG. Reader sends $N$ to tag and $M$ to backend server. Here, ‘modulo’ operator can be used on $N_R$ to make it less complex for calculating $N_R^{th}$ pseudorandom number ($M$). The same must be done at tag side.
(5) To confirm the reader is authenticated, the tag calculates $M' = N_R^{th}$ pseudorandom number by taking $S$ (secret stored in tag) as initial seed value. Using $M'$ as seed value, it calculates a random number ($N'$) from PRNG. The tag verifies if $N'$ and $N$ same. If they are same, the tag is confirmed that the information is from the legitimate reader. The tag calculates $U = h(S||M)$ and updates its secret value as $S\oplus U$.

(6) The server after getting $M$ value from reader checks the ‘Flag’ variable.

- If flag = 0, the secret of tag has matched with $S_{new}$ value and calculates $U = h(S_{new}||M)$. Then server updates the secret value in database as $S_{old}= S_{new}$, $S_{new}= S_{new}\oplus U$.
- If flag = 1, the secret of tag has matched with $S_{old}$ value and calculates $U = h (S_{old}||M)$. Then server updates the secret value in database as $S_{new}= S_{old}\oplus U$ and $S_{old}$ remains unchanged.

4. Security analysis

In this section, we give brief security analysis of our scheme against different common possible attacks in an RFID system.

a) Tag Anonymity: Tag secrets are ‘$S$’ and ‘ID’. Our scheme never discloses publicly this information. Getting ‘$S$’ from PRNG ($S\oplus N_R \oplus N_T$) and ID from $h(ID)$ is not possible. From ‘$N’$, an attacker cannot find the secret because $N = PRNG (M)$ and $M = PRNG (S, N_R)$. $M$ is never communicated directly between tag and reader. So, the tag anonymity is satisfied.

b) Replay attack resistance: Getting the messages of a session will not help the attacker to do further operations in next session. On its first communication, the tag calculates $V= PRNG (S\oplus N_R \oplus N_T)$. $N_R$ is a random number received from reader and $N_T$ is generated by the tag. If the attacker somehow knows $N_R$, $N_T$, and $V$, the value of ‘$S$’ is still unknown. Since the values of $N_R$ and $N_T$ change in every session, the replay attack is not possible.

c) De-synchronization resistance: In this attack, the attacker blocks, modifies some messages leading the entities to asynchronous updates. If any of the value {V, H, N_T} communicated from the tag is blocked or changed, then the authentication process will be terminated. If ‘N’ is blocked or modified, the tag will not update ‘$S$’ value. However, in backend server, the secret is updated as $S_{old}= S_{new}$, $S_{new}= S_{new}\oplus U$, if flag =0, and $S_{new}= S_{old}\oplus U$, if flag = 1. So the tag secret will match $S_{old}$ value for the next session and no asynchronous values.

d) Confidentiality and integrity: The secret ‘$S$’ and index pseudonym ‘ID’ are confidential throughout the authentication process. If any message (i.e., V, H, N_T) from the tag is changed, then it will lead to termination on verifying the equality of $h(ID)$ column of the database with H and $V' = PRNG (S_{new}\oplus N_R \oplus N_T)$ or $V'' = PRNG (S_{new}\oplus N_R \oplus N_T)$ with V. Similarly, if N from reader is changed, then the tag can detect by verifying $N'$ with N, where $N' = PRNG(M)$, $M = PRNG (S, N_R)$. Thus, both confidentiality and integrity are ensured.

e) Mutual Authentication: In mutual authentication, one entity authenticates other entity and vice versa. Upon receiving V, H, N_T, N_R from reader, the server checks the existence of H in database and equality of $V'$ or $V''$ with V. The reader authenticates the tag when it receives the secret value $S_{old}$ or $S_{new}$ as ‘$S$’ of the tag from the server. The tag authenticates the reader when the received value of N is equal to the calculated value PRNG (M ), where M = PRNG (S, N_R).

f) Traceability resistance: In our scheme, the messages communicated between the reader and tag (i.e., V, N) change at each session because V= PRNG (S\oplus N_R \oplus N_T) and N= PRNG (PRNG (seed, N_R)), where $N_R$ and $N_T$ are random numbers generated for every session. The secret value is also updated on every successful completion of a session. Hence, it is very difficult for an adversary to track a tag based on the eavesdropped messages.

g) Forward secrecy: If an attacker gets access to the secret information about the tag in present session i.e. $S_i$, the attacker will not be able to get the secret of previous session $S_{i-1}$. Having value of $S_i$, ID, and communicated messages in session i-1 will not reveal $S_{i-1}$ because $S_i = S_{i-1}\oplus U$, where $U = h (S_{i-1}||M)$ and $M'$ is itself calculated using the $S_{i-1}$ value. The attacker can only get information whether the previous session was successful or not, but getting $S_{i-1}$ from $S_i$ is not possible. Thus, it satisfies forward security.

h) Man-In-The-Middle (MITM) Attack resistance: The attacker can never be able to get meaningful information ($S$, ID) from the communicated messages $N_R$, V, H, N_T, and N because H is calculated as h(ID) using a one-way hash function. The values V, N are computed using PRNG operation with secret ‘$S$’ as seed. If the attacker changes
any message, it will lead to integrity problem and while checking the equality between the received value and the calculated value, due to inequality, the authentication process will be stopped.

**i) DoS Attack resistance:** In this type of attack, the attacker blocks some messages that lead to asynchronous updates in database value and tag secret value. If it happens, then next session onwards the legitimate tag will not be authenticated due to the value mismatch. In our scheme, the database of backend server contains $S_{\text{New}}$ and $S_{\text{Old}}$ to resist this attack. If the message is blocked by the attacker while the reader sends ‘N’ to the tag, then the $S$ value cannot be updated. But next time the tag can communicate with its value as the $S_{\text{Old}}$ value already exists in database. In success termination, the $S_{\text{New}}$ value of database will be used as the tag secret otherwise $S_{\text{Old}}$ will be used. In this way, our protocol can stand against the DoS attack.

In Table 2, we analyse security aspects of our scheme along with some recent authentication schemes against the most commonly occurring attacks in RFID systems.

<table>
<thead>
<tr>
<th></th>
<th>Fu et al. (^5)</th>
<th>Gui et al. (^9)</th>
<th>Chang et al. (^11)</th>
<th>Li et al. (^13)</th>
<th>Min et al. (^7)</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tag Anonymity</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Replay attack resistance</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>De synchronization resistance</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Confidentiality and integrity</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Mutual Authentication</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
<tr>
<td>Traceability resistance</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>Forward secrecy</td>
<td>N</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>MITM Attack resistance</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>DoS Attack resistance</td>
<td>N</td>
<td>N</td>
<td>N</td>
<td>Y</td>
<td>N</td>
<td>Y</td>
</tr>
</tbody>
</table>

Y: Satisfies the property, N: Does not satisfy the property.

5. Performance analysis

Here, we discuss performance analysis of our proposed scheme in terms of operations involved, storage, and communication overhead. The cost of operations of different protocols/schemes along with our scheme is shown in Table 3. The number of communications made with the tag is also mentioned in this table.

<table>
<thead>
<tr>
<th></th>
<th>No. of communications with tag</th>
<th>Computation cost at tag</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fu et al. (^5)</td>
<td>4</td>
<td>$5T_C + 2T_X + 4T_H + 2T_R$</td>
</tr>
<tr>
<td>Gui et al. (^9)</td>
<td>4</td>
<td>$3T_C + 2T_X + 3T_H + 2T_R + 1T_F$</td>
</tr>
<tr>
<td>Chang et al. (^11)</td>
<td>3</td>
<td>$5T_C + 11T_X + 4T_H + 2T_R$</td>
</tr>
<tr>
<td>Li et al. (^13)</td>
<td>3</td>
<td>$2T_C + 2T_X + 4T_H + 1T_R$</td>
</tr>
<tr>
<td>Min et al. (^7)</td>
<td>3</td>
<td>$(a^m + b^n)T_X + 2T_H + 2T_R + 1T_F$</td>
</tr>
<tr>
<td>Ours</td>
<td>3</td>
<td>$T_C + 1T_X + 2T_H + 1T_R + 3T_F$</td>
</tr>
</tbody>
</table>

Here we have denoted the symbols as follows:

- $T_C$: Concatenation cost,
- $T_X$: XOR cost,
- $T_H$: Hash function cost,
- $T_R$: Random number generation cost,
- $T_F$: Flip operation cost,
- $T_S$: Circular shift cost,
- $T_P$: PRNG operation cost.
For Min et al.’s protocol\(^7\), \(m\) and \(n\) are lengths of two base arrays that store \(a\)-bit base tokens and \(b\)-bit base indicators. Doing so requires more storage spaces in tag.

In tag, we store only two types of information i.e., secret of tag (S) and ID pseudonym. According to EPC global GEN2 standard of RFID, we have chosen moderate value of 128 bits as tag’s secret and 96 bits of ID. Thus, total 224 bits of storage requirement out of which 128 bits are programmable as the secret is updated after every session. As the recent standard passive tags contain memory in Kilobits\(^{15}\) (Tego UHF passive tag can store 2Kb, so 11% memory used for our scheme), we only use a small part of it and the remaining can be used for computations.

6. Conclusion

In this paper, we have proposed a secure authentication scheme that can be implemented in low cost passive tags. The recent passive tags (like EPC-C1-GEN2 standard) come with on-chip pseudorandom generator (PRNG). So, the operations involved in our scheme can be implemented with low cost. As the PRNG takes secret of 128 bit as seed value along with random numbers each time; it also satisfies the randomness property for the RFID tag. Our scheme is secure against the common attacks possible in an RFID system and can be extended to different areas like product authentication system, vehicle tracking, wireless sensor networks, mobile phone based tracking systems, supply chain system, etc.

References: