Weaknesses in a Recently Proposed RFID Authentication Protocol

Mete Akgün[1], and M. Ufuk Çağlayan[2]

[1]Tübitak UEKAE, 41470, Kocaeli, Turkey
mete.akgun@tubitak.gov.tr
[2]Computer Engineering Department, Boğaziçi University İstanbul, Turkey
caglayan@boun.edu.tr

Abstract. Many RFID authentication protocols have been proposed to provide desired security and privacy level for RFID systems. Almost all of these protocols are based symmetric cryptography because of the limited resources of RFID tags. Recently Cheng et. al have been proposed an RFID security protocol based on chaotic maps. In this paper, we analyze the security of this protocol and discover its vulnerabilities. We firstly present a de-synchronization attack in which a passive adversary makes the shared secrets out-of-synchronization by eavesdropping just one protocol session. We secondly present a secret disclosure attack in which a passive adversary extracts secrets of a tag by eavesdropping just one protocol session. An adversary having the secrets of the tag can launch some other attacks.

Key words: RFID, Authentication, Security, Privacy

1 Introduction

Radio Frequency Identification (RFID) technology utilizes radio frequency in order to remotely identify people or objects. RFID systems typically consists of three elements: tags, readers and a back-end server. Many people in the world are aware of the benefits of this technology. However, these people have concerns about security and privacy problems of this technology. In the past, many authentication protocols have been proposed in order to provide adequate security and privacy level. However, many studies showed that authentication protocols that are suitable for low-cost RFID tags have serious security and privacy vulnerabilities.

RFID systems have some weak features in terms of security and privacy. These features are an insecure wireless communication between the tag and the reader, accessibility of tags by any reader and tampering tags. Furthermore, RFID tags are not powerful devices in terms of storage and computation capability. Therefore, researchers must consider not only security and privacy threats but also storage and computation capabilities of RFID tags when designing an RFID authentication protocol.

Recently, an RFID authentication protocol has been proposed by Cheng et al. [2]. It is claimed that the proposed protocol provides almost all security properties in the literature. Nevertheless we show that their proposal has security weakness against de-synchronization attacks. We also present an attack that can disclose the secrets of a tag. An adversary can launch some other attacks by using these extracted secrets. The success probabilities of the proposed attacks are significant and their complexities are polynomial.

The rest of this paper is organized as follows. In Section 2, we review the related work. Some preliminaries are introduced in Section 3. We describe Cheng et al.'s authentication protocol in Section 4 and analyze its vulnerabilities in Section 5. At last, we conclude the paper.

2 Preliminaries

In this section, we give the definition and properties of a Chebyshev chaotic map. The fundamental introduction was proposed by Wang and Zhao [3].

Definition 1 (Chebyshev polynomials [2]). *Let n be an integer, and x can be defined as a variable value over the interval [1,1]. Chebyshev polynomial maps $T_n : R \to R$ of degree n is derived from the following recurrent function:*

$$T_n(x) = 2xT_{n-1}(x) - T_{n-2}(x) \tag{1}$$

where the integer $n \geq 2$, $T_0(x) = 1$, and $T_1(x) = x$.

Remark 1. The Chebyshev polynomial should be restricted to the interval [-1,1] so that the action of the map $T_n : T_n([-1, 1]) \to [-1, 1]$ is is characteristic for all $n > 1$. It satisfies a unique, absolutely continuous, invariant measure with positive Lyapunov exponent ($\ln n$) and the Chebyshev map reduces to the feature logistic map for $n > 2$.

Definition 2. *Let n be an integer, and x can be defined as a variable value over the interval [1,1]. The Chebyshev polynomial $T_n(x) : [-1, 1] \to [-1, 1]$ is defined as:*

$$T_n(x) = cos(n.arccos(x)) \tag{2}$$

Definition 3 (Semi-group property). *The Chebyshev polynomial exhibits a well-known property, so-called the semi-group property, which presents that*

$$T_r(T_s(x)) = T_{r.s}(x) \tag{3}$$

Definition 4. *Commute under composition. An immediate consequence of the semi-group property is that Chebyshev polynomials commute under composition:*

$$T_r(T_s(x)) = T_s(T_t(x)) \tag{4}$$

Definition 5 (Enhanced Chebyshev polynomials). *The enhanced Chebyshev polynomials establish that*

$$T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x)) \bmod N \tag{5}$$

where the integer $n \geq 2$, $x \in (-\infty, +\infty)$, *and* N *is a large prime number. It absolutely derives the following relation:*

$$T_{r.s} = T_r(T_s(x)) = T_s(T_t(x)) \tag{6}$$

Thus, the semi-group property still can be achieved, and the enhanced Chebyshev polynomials also commute under composition.

3 Cheng et al.'s Protocol

In 2013, Cheng et al. proposed an RFID mutual authentication protocol based on chaotic maps [2]. They utilized enhanced Chebyshev polynomials in the proposed protocol (Definition 5). The proposed protocol needs seven exclusive-or and two chaotic cryptographic operations on the tag side. The authors presented the authentication proof of the proposed protocol based on Burrows-Abadi-Needham logic [1]. They also claim that their protocol provides the following security requirements: resistance to replay attacks, resistance to impersonation attacks, resistance to denial-of-service attacks, location privacy and forward secrecy.

3.1 Protocol Description

We give the overview of Cheng et al. protocol in Figure 1 and notations are listed in Table 1.

Table 1. Notations for Cheng et al.'s Protocol

Notation	Description
ID	The secure identity of the tag
$H(ID)$	The hash value of the identity of the tag
x	The current session key
x_{old}	The last successfully verified session key
$H(ID) \oplus x$	The value used as an index to query the database
$T_{.}(.)$	The enhanced Chebyshev polynomial
$\oplus$	The bit-wise XOR operation
$\in$	Random choice operator
$\leftarrow$	The substitution operation

For each tag T, the back-end server stores the following entry: $[H(ID) \oplus x_{old}, H(ID) \oplus x, H(ID), ID, x, x_{old}]$. The tag T stores the current session key

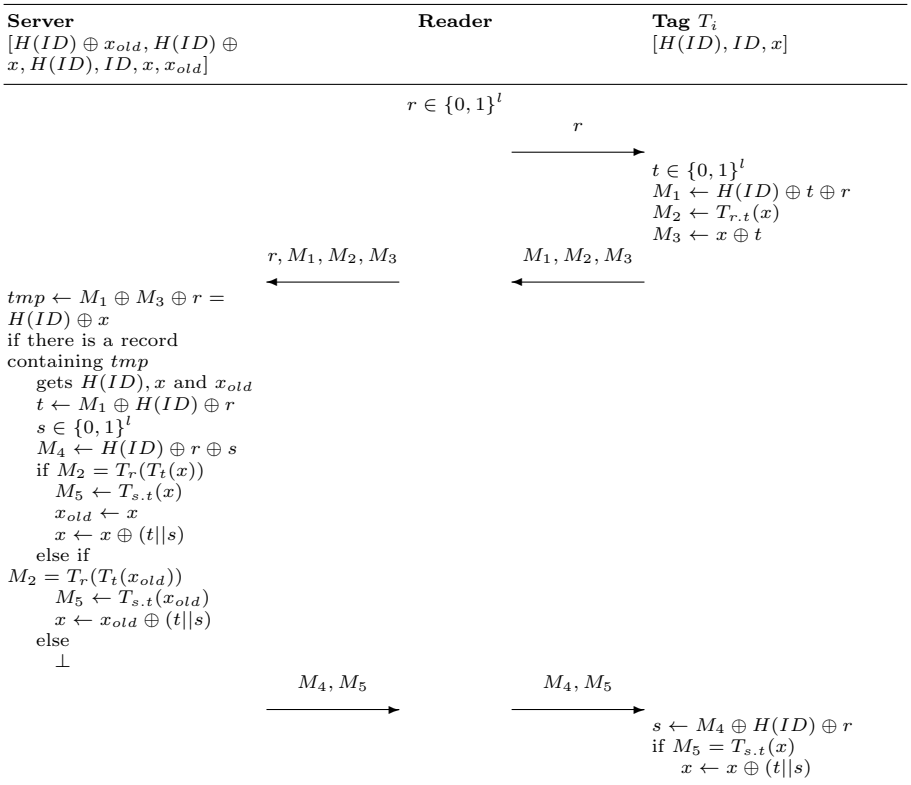

Fig. 1. Cheng et al.'s Protocol

x, the secure identity ID and the hashed value of secure identity $H(ID)$. It is assumed that $x_{old} = x$ initially. A step by step description of Cheng et al.'s Protocol is given below

1. The reader generates a random number r and sends it to the tag.
2. Upon receiving the random number r, the tag generates a random number t and computes $M_1 \leftarrow H(ID) \oplus t \oplus r$,$M_2 \leftarrow T_r(T_t(x))$ and $M_3 \leftarrow x \oplus t$. Then, the tag sends M_1, M_2 and M_3 to the reader.
3. After receiving the messages from the tag, the reader forwards them with the random number r to the back-end server.
4. After receiving the messages from the reader, the back-end server computes $H(ID) \oplus x = M_1 \oplus M_3 \oplus r$. The back-end server checks if there is a record matching with the index $H(ID) \oplus x$. If it finds a record, it gets $H(ID)$, x and x_{old}. Then, it computes $t \leftarrow M_1 \oplus H(ID) \oplus r$ and checks the validity of M_2 by computing $T_r(T_t(x))$ and $T_r(T_t(x_{old}))$. If M_2 is valid, the back-end server generates a random number s and computes $M_4 \leftarrow H(ID) \oplus r \oplus s$, otherwise the session is stopped. If $M_2 = T_r(T_t(x))$, the server computes

$M_5 = T_s(T_t(x))$ and replaces x and x_{old} with $x \oplus (t||s)$ and x respectively. If $M_2 = T_r(T_t(x_{old}))$, the server computes $M_5 = T_s(T_t(x_{old}))$ and replaces x with $x_{old} \oplus (t||s)$. The server sends M_4 and M_5 to the reader.

5. After receiving the messages from the back-end server, the reader forwards them to the tag.
6. After receiving the messages from the reader, the tag computes $s \leftarrow M_4 \oplus H(ID) \oplus r$ and checks the validity of M_5 by computing $T_s(T_t(x))$. If M_5 is valid, it replaces x with $x \oplus (t||s)$.

3.2 Security Properties

Cheng et al.'s protocol is asserted to have a list of security properties. These properties provided in [2] are summarized below.

- **Mutual Authentication:** Mutual authentication is proved by using Burrows-Abadi-Needham (BAN) logic proof [1].
- **Secrecy protection:** Any secret data cannot be retrieved by any attacker from the communications between the tag and the back-end server. The secret value x is well protected by the enhanced Chebyshev polynomial.
- **Resistance to impersonation attack:** Without knowing the random value t selected by the legal tag and the secret value x stored in the memory of the tag, an attacker cannot pass the authentication in the server side. Only the valid server can compute the correct values M_4 and M_5 with its own selected random number so the attacker cannot pass the tag's authentication.
- **Resistance to replay attack** It is impossible to intercept messages with the intention of replaying them, since any message or information sent from the three components (tag, reader, and server) can always be changed by using random numbers t, r, and s. The random numbers t and s are transmitted securely by using the enhanced Chebyshev polynomials.
- **Resistance to denial-of-service attack** Although the synchronous updating is thus interrupted, the tag's original secret value still can match x_{old} to pass the authentication, such that $M_2 = T_r(T_t(x_{old}))$.
- **Location privacy** Random values t and s that are randomly selected by the tag and the server, respectively, are used to generate the essential data M_2 and M_5 and are used to update the secret constantly. r, t, and s values make the communication messages unpredictable for attackers.
- **Forward secrecy** Even if the attacker has the ability to compromise current session negotiations and retrieve the secret value, he or she still cannot use the compromised data to derive details of previous communications. This is because each session has a different secret x, and the shared key is always updated after individual tag reading.

4 Attacks

4.1 De-synchronization Attack

We present an attack in which a passive adversary impersonates the tag to the back-end server without knowing tag's secrets. At the end of the attack, the

back-end server performs key-updating but the tag does not. Therefore, the synchronization of the session key between the tag and the back-end server is broken. The details of this attack are given below:

We know that the back-end server has two registers for x values corresponding to the attacked tag namely: x^s_{old} and x^s_{new}. The tags has a register for the current value of x namely: x^t. At the beginning of the attack, the content of the registers are shown in Table 2.

Table 2. The content of the registers at the beginning of the attack.

Register	Value
x^s_{new}	x
x^s_{old}	x
x^t	x

Phase 1:

1. An adversary queries a tag T with a number $r^1 = 1$.
2. After receiving the number r^1, the tag T computes $M^1_1 \leftarrow H(ID) \oplus t^1 \oplus r^1$,$M^1_2 \leftarrow T_{r^1}(T_{t^1}(x))$ and $M^1_3 \leftarrow x \oplus t^1$ and sends them to the adversary.
3. The adversary computes $H(ID) \oplus t^1 \leftarrow M^1_1 \oplus r^1$. She knows M^1_2 equals $T_{t^1}(x)$ because r_1 equals to 1 (Definition 1).

At the end of the Phase 1, neither the tag nor the back-end server performs key-updating. The content of the registers are shown in Table 2.

Phase 2:

1. The reader initiates a valid session by querying tags with a random number r^2.
2. After receiving the random number r^2, the tag T computes $M^2_1 \leftarrow H(ID) \oplus t^2 \oplus r^2$,$M^2_2 \leftarrow T_{r^2}(T_{t^2}(x))$ and $M^2_3 \leftarrow x \oplus t^2$ and sends them to the reader.
3. The reader forwards r^2, M^2_1, M^2_2 and M^2_3 to the back-end server.
4. The server identifies the tag T. It computes $M^2_4 \leftarrow H(ID) \oplus r^2 \oplus s^2$ and $M^2_5 \leftarrow T_{s^2.t^2}(x)$ and sends them to the reader.
5. The reader forwards M^2_4 and M^2_5 to the tag.
6. At the end of this valid session, the tag and the back-end server perform key-updating.

At the end of the Phase 2, the content of the registers are as shown in Table 3.

Phase 3:

1. The reader initiates a valid session by querying tags with a random number r^3.

Table 3. The content of the registers at the end of Phase 2.

Register	Value
x^s_{new}	$x \oplus (t^2 \| \| s^2)$
x^s_{old}	x
x^t	$x \oplus (t^2 \| \| s^2)$

2. After receiving the random number r^3, the adversary has to create valid messages in order to pass the check by the back-end server. She obtained $H(ID) \oplus t^1$, $T_{t^1}(x)$ and $x \oplus t^1$ in the Phase 1. She will use the values to create valid M_1^3, M_2^3 and M_3^3. She computes $M_1^3 \leftarrow H(ID) \oplus t^1 \oplus r^3$, $M_2^3 \leftarrow T_{r^3}(T_{t^1}(x))$ and $M_3^3 \leftarrow x \oplus t^3$ and sends them to the adversary.
3. The reader forwards r^3, M_1^3, M_2^3 and M_3^3 to the back-end server.
4. The back-end server computes $H(ID) \oplus x = M_1^3 \oplus M_3^3 \oplus r^3$. The back-end server gets $H(ID)$ and x^s_{old} from the record matching with the index $H(ID) \oplus x$. We know that the content of the register x^s_{old} equals x. The back-end server computes $t^1 \leftarrow M_1^3 \oplus H(ID) \oplus r^3$. It checks the validity of M_2^3 by computing $T_{r^3}(T_{t^1}(x))$. The adversary passes this check because she creates M_2^3 with the valid r^3 and t^1 values. After that the back-end server generates a random number s^3 and replaces x^s_{new} and x^s_{old} with $x \oplus (t^1 \| \| s^3)$ and x respectively.

At the end of the Phase 3, the content of the registers are as shown in Table 4. In the above attack, the adversary is authenticated by the back-end database as a legitimate tag with a success probability of 1. The given attack makes the shared secrets out-of-synchronization in which only one legal protocol session is required.

Table 4. The content of the registers at the end of Phase 2.

Register	Value
x^s_{new}	$x \oplus (t^1 \| \| s^3)$
x^s_{old}	x
x^t	$x \oplus (t^2 \| \| s^2)$

4.2 Secret Disclosure Attack

In this section, we present a passive attack in which an adversary retrieves secret information $H(ID)$ and x in the tag. In this attack, an adversary benefits from

weakness in key-updating mechanism. She can disclose all secret parameters by eavesdropping one session of the protocol as follows:

1. An adversary eavesdrops a transcript of one protocol session between the tag T and the reader. She stores r, M_1, M_2 and M_3.
2. The adversary queries the tag T with the random number r'.
3. After receiving r', the tag T computes M_1', M_2' and M_3' and sends them to the adversary.
4. The adversary computes $(M_1' \oplus M_3' \oplus r') \oplus (M_1 \oplus M_3 \oplus r) = (H(ID) \oplus x') \oplus (H(ID) \oplus x) = x' \oplus x = x \oplus (t||s) \oplus x = (t||s)$. The adversary gets the values of t and s. She computes $M_1 \oplus r \oplus t = H(ID)$ and $M_3 \oplus t = x$. The adversary gets the values of $H(ID)$ and x. Finally, she computes $x \oplus (t||s) = x'$.

An adversary knowing the secret values $H(ID)$ and x' can easily perform traceability, tag impersonation, reader impersonation and de-synchronization attacks with a success probability 1.

5 Conclusion

In this paper, we show that Cheng et al.'s protocol [2] suffers from semi-group property of Chebyshev polynomials and its weak key-updating mechanism. We discover that this protocol is vulnerable to de-synchronization attack and secret disclosure attack. The cost of our attacks is the eavesdropping of one protocol session. The proposed secret disclosure attack shows that no security or privacy properties are achieved by this protocol.

References

1. Burrows, M., Abadi, M., Needham, R.: A Logic of Authentication. ACM Trans. Comput. Syst. **8** (1990) 18–36
2. Cheng, Z.Y., Liu, Y., Chang, C.C., Chang, S.C.: Authenticated RFID security mechanism based on chaotic maps. Security and Communication Networks **6** (2013) 247–256
3. Wang, X., Zhao, J.: An improved key agreement protocol based on chaos. Communications in Nonlinear Science and Numerical Simulation **15** (2010) 4052 – 4057