Journal of Wireless Personal Communications manuscript No.
(will be inserted by the editor)

Security attacks and enhancements to chaotic map-based RFID authentication protocols

Süleyman Kardaş · Ziya Alper Genç

21/01/2017 / Revision: date

Abstract Radio Frequency IDentification (RFID) technology has been increasingly integrated into numerous applications for authentication of objects or individuals. However, because of its limited computation power, RFID technology may cause several security and privacy issues such as tracking the owner of the tag, cloning of the tags and etc. Recently, two chaotic map-based authentication protocols have been proposed for low-cost RFID tags in order to eliminate these issues. In this paper, we give the security analysis of these protocols and uncover their weaknesses. We prove that these protocols are vulnerable to tag tracing, tag impersonation and de-synchronization attacks. The attack complexity of an adversary is polynomial and the success probability of these attacks are substantial. Moreover, we also propose an improved RFID authentication protocol that employs Chebyshev chaotic maps and complies with the EPCglobal Class 1 Generation 2 standard. Finally, we show that our protocol is resistant against those security issues.

Keywords RFID · Security · EPC Global · Chaotic Map

1 Introduction

Radio Frequency Identification (RFID) technology have been used for automated remote identification of objects or people by means of small, lightweight and inexpensive RFID tags. By virtue of the ease of deployment and low cost,

The research was carried out while the second author was with TUBITAK BILGEM.

Corresponding author: S. Kardas
Batman University, Faculty of Engineering and Architecture, Batman, Turkey
Tel.: +905079874354
E-mail: skardas@gmail.com

Z. A. Genç
University of Luxembourg, Luxembourg
E-mail: ziya.genc@uni.lu

this technology has been broadly deployed to several daily-life applications; for instance, in logistics, asset tracking, warehouse management, library books check-in/check-out, medicine, embedded cards [24]. However, due to the nature of the communication between readers and tags which works on insecure wireless channel, such broad adoption of RFID technology raises important security and privacy issues. For this reason, the authentication protocols used in RFID technology should be seriously taken into account in order to protect the privacy of tag owner [25].

A simple RFID system is composed of at least one reader (transceiver), several RFID tags/labels (transponders) with limited energy consumption and a back-end server that stores a database. Each tag is enclosed to a particular object and has a unique identity in order to be tracked easily. Chien [12] categorizes RFID tags into four different types: ultralight, lightweight, simple, and high cost tags. Each type has different computational capabilities. For instance, RFID tags in ultralight category may perform only bitwise, XOR, AND and OR operations whereas the ones in the high cost category may support public-key cryptography. Second set of RFID components are the readers where each reader commonly consists of a control unit, an RF module, and a coupling element in order to query tags/labels by means of RF communication. The last component is the back-end server which generally stores the information about the tags in its database.

The restriction on tag's computational capability affects on the targeted level of security and privacy of RFID systems. On the other hand, when the security of an RFID system is not designed carefully, the system can be vulnerable to a wide range of attacks such as replay attack, impersonation attack, tracking attack and denial-of-service (DoS) attack. Namely, the more restrictions on the computational resources, it is more challenging to design a secure privacy preserving authentication protocol. In the literature, these challenges encourage researchers to propose several authentication schemes [12,27,4,9, 21,29,23,18,30]. Many of these solutions are very complex and requires computations with high costs and are not compatible for ultralight RFID systems. Besides, diverse number of ultralight authentication protocols have been recently published in the literature [13,35,10,14] but most of them have security and privacy weaknesses and do not achieve the targeted security and privacy levels.

Moreover, considering only ultralight RFID systems, the EPCglobal Class-1 Generation-2 (C1G2) standard has been published in order to provide a universal model for tag and reader communications [1]. This standard has possessed low implementation cost and high performance in order to serve for tag singularization and chain applications. Since the standard aims only to achieve identification performance, it lacks security and privacy features. Even eavesdropping the messages between the reader and the tag, an adversary can discover the plain messages. Notwithstanding, Cheng *et al.* recently proposed a lightweight RFID mutual authentication protocol [11], based on chaotic maps [33] where the enhanced Chebyshev polynomials have been utilized. Benssalah *et al.* [6] proved that Cheng's *et al.*'s solution has security weaknesses on shared secret updating and message generation. It is also shown in [3] that this protocol is vulnerable to secret disclosure attack and desynchronization attack. Then, Benssalah *et al.* [6] also proposed an enhanced secure RFID authentication protocol that uses Chebyshev chaotic maps as the underlying hard problem. This protocol conforms to the EPC C1-G2 standard with more flexibility and mobility for RFID applications. However, in [2], it is figured out that Benssalah *et al.*'s protocol is also vulnerable to tracking, tag impersonation, and de-synchronization attacks. Akgun *et al.* also proposed another chaotic map-based RFID authentication protocol which is compatible with EPC C1 G2 standard.

Our contribution: In this paper, we first define a simple generic attack that can be applied to any RFID authentication protocol. We applied this attack to Benssalah *et al.*'s authentication protocol and we observed that their protocol does not provide resistance against tag tracking. Next, we analyzed Akgun *et al.*'s protocol and proved that their protocol does not satisfy security claims against our attack. Finally, we propose a new enhanced RFID authentication protocol and show that this protocol eliminate those security threats.

Roadmap: The outline of the paper is given as follows. In Section 2, we give the preliminaries about Chebyshev polynomials and the security and privacy definitions. Section 3 describes Benssalah *et al.*'s protocol and then describes Akgun *et al.*'s authentication protocol. Section 4 introduces our proposed privacy preserving authentication protocol. In Section 5, we first give the attacks on Benssalah *et al.*'s protocol and Akgun *et al.*'s protocol. Then, we give the security proof of our protocol. Section 6 gives the performance comparison of our protocol with the existing protocols. Finally, Section 7 concludes the paper.

2 Preliminaries

In this section, we first review the definitions of Chebyshev polynomial and the hard problems that employ the enhanced Chebyshev polynomials. Then, we provide the security and privacy definitions which will be used throughout the paper.

2.1 Chebyshev Polynomials

In this section, we borrow the definitions from [11], [36] and [37]. The formal definition of Chebyshev chaotic maps that are proposed by Wang and Zho [32] are given as follows.

Definition 1 (Chebyshev polynomials) Let $x \in [-1, 1]$, and $n \in \mathbb{N}$. Then, the subsequent function describes a Chebyshev polynomial map $T_n : R \to R$ of degree n:

$$T_n(x) = cos(nxarccosx), \{x| - 1 \leq x \leq 1\},$$

and the recurrent formulas are

$$Tn(x) = 2xT_{n-1}(x) - T_{n-2}(x)$$

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

Using Definition 1, we have the following first few polynomials: $T_2(x) = 2x^2 - 1, T_3(x) = 4x^3 - 3x$, and $T_4(x) = 8x^4 - 8x^2 + 1$. Moreover, Chebyshev polynomials have the following semi-group and commutativity property.

Definition 2 (Semi-group property) Chebyshev polynomials have the following semi-group property:

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

Definition 3 (Commutativity property) Chebyshev polynomials have the following commutativity property:

$$T_r(T_s(x)) = T_s(T_r(x))$$

Based on Definition 1, the improved Chebyshev polynomials can defined as follows. Note that this definition has been employed in the protocol designs.

Definition 4 (Improved Chebyshev Polynomials) Let $x \in [-\infty, +\infty]$, N be a big large number and $n \geq 2$. Then, the following function defines an improved version of Chebyshev polynomial:

$$T_n(x) = (2xT_{n-1}(x) - T_{n-2}(x)) \; mod \; N$$

Obviously,

$$T_r(T_s(x)) = T_s(T_r(x)) = T_{rs}(x)$$

Hence the semigroup property still holds and the improved Chebyshev polynomials also commute under composition.

Definition 5 (Discrete Logarithm Problem) DLP is described as follows: Given values of x and y, find an α such that $T_\alpha(x) = y$.

Definition 6 (Diffie–Hellman Problem) DHP is described as follows. Given a values of x, $T_s(x)$ and $T_r(x)$, what is values of $T_{rs}(x)$?

2.2 The Security Definitions

RFID authentication systems are always under the threat of man-in-the-middle attacks due to easy eavesdropping the air transmission between tags and reader. During transmission, an adversary is able to listen, modify or block the messages being transmitted. He/she can also retransmit the messages maliciously to interrogate the tag or the back-end server in order to impersonate the reader or the victim tag. Several security and privacy issues and adversarial models are addressed in [22,19,5,31,15] in details. In this paper, we consider the following security and privacy notions in the security analysis of authentication protocols.

Paise *et al.* defines a secure authentication protocol as follows [28].

Definition 7 (Security) An authentication protocol achieves *security* provided that it performs both secure tag authentication and secure reader authentication. Let $\mathcal{A}$ be a polynomial-time bounded adversary, $\mathcal{T}_i$ be an uncorrupted alive legal tag and $\mathcal{R}$ be the legitimate reader. Then,

- if $\mathcal{R}$ identifies $\mathcal{A}$ as $\mathcal{T}_i$ on the session π with non-negligible probability and there is not any matching conversation between $\mathcal{T}_i$ and π, then tag authentication is not secure.
- if $\mathcal{T}_i$ identifies $\mathcal{A}$ on the session π with non-negligible probability and there is not any matching conversation between $\mathcal{R}$ and π, then reader authentication is not secure.

Avoine defines the privacy notion of universal untraceability as follows [5].

Definition 8 (Universal untraceability) In universal untraceability, an adversary cannot correlate two responses of a tag, where the responses are disconnected with the help of at least a single successful authentication. The following universal untraceability experiment is played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

- *Learning Phase*: Let $\mathcal{A}$ communicates with two legal tags $\mathcal{T}_0$ and $\mathcal{T}_1$. $\mathcal{A}$ is capable of initiating, monitoring, injecting any message, and stopping the sessions between $\mathcal{T}_0$ and $\mathcal{R}$. $\mathcal{A}$ can also start, monitor or stop the sessions between $\mathcal{T}_1$ and $\mathcal{R}$.
- *Challenge Phase*: Let a challenger $\mathcal{C}$ allow the reader to run successful protocol transcripts with $\mathcal{T}_0$ and $\mathcal{T}_1$. Then, $\mathcal{C}$ randomly selects one of them as a target tag, $\mathcal{T}_b$. $\mathcal{A}$ communicates with $\mathcal{T}_b$. Besides, $\mathcal{A}$ can also initiate, monitor or stop the authentication sessions between $\mathcal{T}_b$ and $\mathcal{R}$.
- *Guessing Phase*: Finally, $\mathcal{A}$ finishes the experimental game and produces an output bit b'.

Avoine also introduces the privacy notion of existential untraceability as follows [5].

Definition 9 (Existential untraceability) In existential untraceability, an adversary cannot correlate two responses of a given tag, where these responses are not necessarily isolated by a successful authentication. The following existential untraceability experiment is played between a challenger $\mathcal{C}$ and an adversary $\mathcal{A}$.

- *Learning Phase*: Let $\mathcal{A}$ communicates with two legitimate tags $\mathcal{T}_0$ and $\mathcal{T}_1$. $\mathcal{A}$ is capable of initiating, monitoring, injecting any message or stopping the sessions between $\mathcal{T}_0$ and $\mathcal{R}$. $\mathcal{A}$ can also start, monitor or stop the sessions between $\mathcal{T}_1$ and $\mathcal{R}$.
- *Challenge Phase*: $\mathcal{C}$ randomly selects one of the legitimate tags as a target tag, $\mathcal{T}_b$. $\mathcal{A}$ communicates with $\mathcal{T}_b$. $\mathcal{A}$ can also initiate, monitor or stop authentication sessions between $\mathcal{T}_b$ and $\mathcal{R}$.
- *Guessing Phase*: Finally, $\mathcal{A}$ finishes the experimental game and produces an output bit b'.

Furthermore, a strong adversary can corrupt $\mathcal{T}_i$ at time t and access to the whole secret values of the tag. Forward untraceability is described as no RFID tag can be traced with the help of past transactions of the tag with the reader even though the adversary tampered the tag [27]. The notion of forward privacy formally described in [7] as follows.

Definition 10 (Forward privacy) In forward privacy, the following experiment is played between a challenger $\mathcal{C}$ and an strong adversary $\mathcal{A}$.

- *Learning Phase*: Let two legitimate tags $\mathcal{T}_0$ and $\mathcal{T}_1$ communicate with $\mathcal{A}$. $\mathcal{A}$ is capable of initiating, monitoring, injecting any message, and dropping the sessions between $\mathcal{T}_0$ and $\mathcal{R}$. $\mathcal{A}$ can also start, monitor, and drop the sessions between $\mathcal{T}_1$ and $\mathcal{R}$.
- *Challenge Phase*: Let a challenger $\mathcal{C}$ allow the reader to run successful protocol transcripts with $\mathcal{T}_0$ and $\mathcal{T}_1$. $\mathcal{C}$ randomly selects one of them as a target tag, $\mathcal{T}_b$. $\mathcal{A}$ communicates with $\mathcal{T}_b$. $\mathcal{A}$ can initiate, monitor, and drop authentication sessions between $\mathcal{T}_b$ and $\mathcal{R}$. $\mathcal{A}$ has also access to authentication outcomes and internal state of $\mathcal{T}_b$.
- *Guessing Phase*: Finally, $\mathcal{A}$ finishes the experimental game and produces an output bit b'.

3 Previous Art on Chaotic Map Based Authentication Protocol

In this section, we give the detailed descriptions of two RFID authentication protocols, which utilize the chaotic maps as the underlying hard problem. The first protocol is Benssalah *et al.*'s authentication protocol. The latter one is Akgun *et al.*'s authentication protocol which is the improved version of the former one. Both protocols essentially uses cryptographic primitives such as chaotic maps, hash function and xor operation at the reader side but uses only chaotic map and xor in the tag side.

3.1 Benssalah *et al.*'s Authentication Protocol

Benssalah *et al.* [6] showed some weaknesses of the protocol of Cheng *et al.* [11] that utilizes the chaotic maps. Then, they suggested an improved version of the protocol and which also employs the enhanced Chebyshev polynomials. Considering the computational requirement, the tags need four chaotic map operations and one pseudo-random number generation. The authors claimed that their authentication scheme is secure against replay, impersonation and denial-of-service attacks. Their protocol also offers mobility and mutual authentication.

3.1.1 Notation

Table 1 provides the notations used throughout the authentication scheme.

Table 1 The following notations are used in [6]

Notation	Definition
ID	ID value of a tag
$h(.), H(.)$	Hash functions
$H(ID)$	Hash of an identity (ID)
x_{new}	The newly generated session key
x_{old}	The old session key
$T.(\cdot)$	The Chebyshev polynomial
$\oplus$	XOR
$\|\|$	Concatenation operator
$\leftarrow$	Substitution operator

3.1.2 Protocol Description

Benssalah *et al.*'s RFID authentication scheme is depicted in Figure 1. The scheme consists of two stages: *initialization* and *authentication*.

The Initialization Stage: For each tag in the system, the server/owner first picks a random secret key x. Next, the server records the credentials of each tag in its own database, which has the following form: $[x_{old}, c_{old}, x_{new}, c_{new}, ID, H(ID)]$ where ID is the tag's identity and c_{old}/c_{new} represents the old and new index values of the tag in the database, respectively. Each tag stores a data in the form of $[x, c_i, ID, H(ID)]$ in its non-volatile memory. RID denotes an identifier of a reader. The initial values are set as $c_{new} \leftarrow c_{old} \leftarrow 0$ and $x_{new} \leftarrow x_{old} \leftarrow x$.

The Authentication Stage: Authentication is carried out by performing the following steps.

1. A legitimate RFID reader first picks a random number $r \epsilon_R \{0,1\}^\ell$ and broadcasts it.
2. Once received r, the tag picks another random number $t \epsilon_R \{0,1\}^\ell$ and computes $M_1 \leftarrow h(ID) \oplus ((r \oplus t)||(t \oplus ID)) \oplus t$, $M_2 \leftarrow T_{r \cdot t}(x)$ and $M_3 \leftarrow x \oplus t$. Then, the tag sends the quadruple (c_i, M_1, M_2, M_3) to the reader.
3. Upon receiving (c_i, M_1, M_2, M_3), the reader gets a time-stamp T and computes $V \leftarrow H(RID \oplus r \oplus T)$. Next, it sends $(r, V, T, c_i, M_1, M_2, M_3)$ to the server.
4. The server verifies the validity of V. If V is valid, the server executes the following transactions depending on the value of c_i.
   (a) If c_i is equal to 0:
      - The server scans the whole database and finds the matched records. For each record in the database, the server computes $T_{old} \leftarrow T_{r \cdot (M_3 \oplus x_{old})}(x_{old})$ and $T_{new} \leftarrow T_{r \cdot (M_3 \oplus x_{old})}(x_{new})$. If M_2 is equal to either T_{old} or T_{new}, the server finds the correct match in the database. Server sets x to x_{old} or x_{new}.
      - The server also checks the validity of the message M_1. If M_1 is not a valid message, it rejects the tag.

(b) If c_i is not equal to 0:
   - c_i is the database index of the current tag. The server gets the record where c_i matches with either c_{old} or c_{new}. Then, it updates $x \leftarrow x_{old}$ or $x \leftarrow x_{new}$.
   - The validity of both M_1 and M_2 are checked by the server . If one of them is invalid, the tag would be rejected.

(c) The server picks another random number $s \epsilon_R \{0,1\}^\ell$ and computes $H_{info} \leftarrow h(data \oplus r)$, $info \leftarrow RID \oplus data$, $M_4 \leftarrow h(ID) \oplus s \oplus r$, $M_5 \leftarrow T_{s \cdot t}(x)$. Then, it sends $(info, H_{info}, M_4, M_5)$ to the reader.

(d) The server also executes $x_{old} \leftarrow x_{new}$, $c_{old} \leftarrow c_{new}$, $x_{new} \leftarrow x_{new} \oplus T_{t||s}(x_{new})$ and $c_{new} \leftarrow T_{s \oplus t}(x_{new})$ for key and index updating.

5. The reader verifies that H_{info} is valid. If H_{info} is valid, the reader sends $(M_4,\ M_5)$ to the tag.
6. The tag recovers s from M_4 by computing $s \leftarrow h(ID) \oplus r \oplus M_4$.
7. The tag also verifies the validity of M_5. If M_5 is valid, the tag executes the following key and index updates $(x \leftarrow x \oplus T_{t||s}(x),\ c_i \leftarrow T_{s \oplus t}(x))$.

3.1.3 Security Analysis of The Protocol

In this protocol, we first give a generic attack that can be applied to any authentication protocol. Then, we show the security issues of this protocol against our attack.

Let $\mathcal{A}$ be an active adversary who can initiate a protocol with a tag or reader, stop the protocol between a tag and reader, modify the responses from a tag or reader.

Definition 11 (Generic Attack) Let $\mathcal{T}$ be the target tag and $\mathcal{R}$ be the legal reader. Our attack consists of two phases.

1. In the first phase, $\mathcal{A}$ modifies the first message of $\mathcal{R}$. Next, the tag computes the response and sends it to $\mathcal{R}$.
2. In the second phase, $\mathcal{R}$ generates its own messages and sends them to the server. However, $\mathcal{A}$ updates the response of the $\mathcal{R}$.

Impersonation Attack: We utilize our generic attack in order to impersonate a victim tag $\mathcal{T}_0$ with another legal tag T_i. In this attack, the reader would authenticate $\mathcal{T}_j$ but the tag in front of $\mathcal{R}$ would be $\mathcal{T}_i$. The attack steps are carried out as follows.

- $\mathcal{R}$ generates a random number r and sends it to the tag $\mathcal{T}_i$ but during the transmission $\mathcal{A}$ corrupts the message and set $r \leftarrow 0$.
- $\mathcal{T}_i$ also generates another random number t and computes M_1, M_2, and M_3. Note that since $r = 0$, $M_2 = T_{0.t}(x) = 1$. $\mathcal{T}_i$ sends the quadruple (M_1, M_2,M_3, c_i) to $\mathcal{R}$.
- After receiving the quadruple, $\mathcal{R}$ generates a time-stamp T and computes $V = H(RID \oplus r \oplus T)$. Next, it dispatches the quadruple along with r, T and V to the server.

Backend Server	**RFID Reader**	**Tag**$_i$
$[x_{old}, c_{old}, x_{new}, c_{new}, ID, H(ID)]$		$[ID, H(ID), x, c_i]$
	Pick a random $r \in_R \{0,1\}^l$	
	$\xrightarrow{r}$	
		Pick a random $t \in_R \{0,1\}^l$ $M_1 \leftarrow H(ID) \oplus$ $((r \oplus t)\|\|(t \oplus ID)) \oplus t$ $M_2 \leftarrow T_{r \cdot t}(x)$, $M_3 \leftarrow x \oplus t$
	$\xleftarrow{c_i, M_1, M_2, M_3}$	
	Generate a timestamp T $V \leftarrow H(RID \oplus r \oplus T)$	
$\xleftarrow{r, V, T, c_i, M_1, M_2, M_3}$		
Pick a random $s \in_R \{0,1\}^l$ $M_4 \leftarrow h(ID) \oplus s \oplus r$ $M_5 \leftarrow T_{s \cdot t}(x)$ $H_{info} \leftarrow H(data \oplus r)$ $info \leftarrow RID \oplus data$		
$\xrightarrow{info, H_{info}, M_4, M_5}$		
	Check if $H_{info} = h(RID \oplus info \oplus r)$	
	$\xrightarrow{M_4, M_5}$	
$x_{old} \leftarrow x_{new}$ $x_{new} \leftarrow x_{new} \oplus T_{s\|\|t}(x_{new})$ $c_{old} \leftarrow c_{new}$ $c_{new} \leftarrow T_{s \oplus t}(x_{new})$		$s \leftarrow H(ID) \oplus r \oplus M_4$ if $M_5 = T_{s \cdot t}(x)$ $x \leftarrow x \oplus T_{t\|\|s}(x)$ $c_i \leftarrow T_{s \oplus t}(x)$

Fig. 1 The protocol of Benssalah *et al.*

- During the transmission $\mathcal{A}$ sets c as 0, $T = T \oplus r$ and $r \leftarrow 0$.
- The server first validates the correctness of V by computing $V' = H(RID \oplus r \oplus T)$. This validation is passed because $RID \oplus 0 \oplus T \oplus r \leftarrow RID \oplus T \oplus r$. Since $c_i = 0$, the server will do full search on the database. For the first element of database, the server computes $T_{old} \leftarrow T_{(M_3 \oplus x_{old}).r}(x_{old})$ and $T_{new} \leftarrow T_{(M_3 \oplus x_{new}).r}(x_{new})$. Note that both T_{old} and T_{new} will be equal to 1 because $r \leftarrow 0$ and the property of Chebyshev polynomial $T_0(x) \rightarrow 1$. Now, the server would see that T_{old} is equal to M_2. The server would also try to verify the correctness of the M_1 but even the server finds that M_1 is not correct, the protocol is not aborted because this case is not really

considered in the protocol. Then, the server computes M_3 and M_4 and sends them to $\mathcal{R}$. Finally, the server updates x_{old}, x_{new}, c_{old} and c_{new}.

After this attack, the reader authenticates first entry of the database but the reader interacts with different tag $\mathcal{T}_i$.

Desynchronization Attack: The protocol is vulnerable to desynchronization attack. When two subsequent impersonation attacks are mounted, x_{old} and x_{new} values are both updated incorrectly. This causes that the first tag in the database can no longer be authenticated by a legal reader.

3.2 Akgun *et al.*'s Authentication Protocol

Akgun *et al.* [2] found some weaknesses on the protocol of Benssalah *et al.* [6] and proposed an improved version of the protocol.

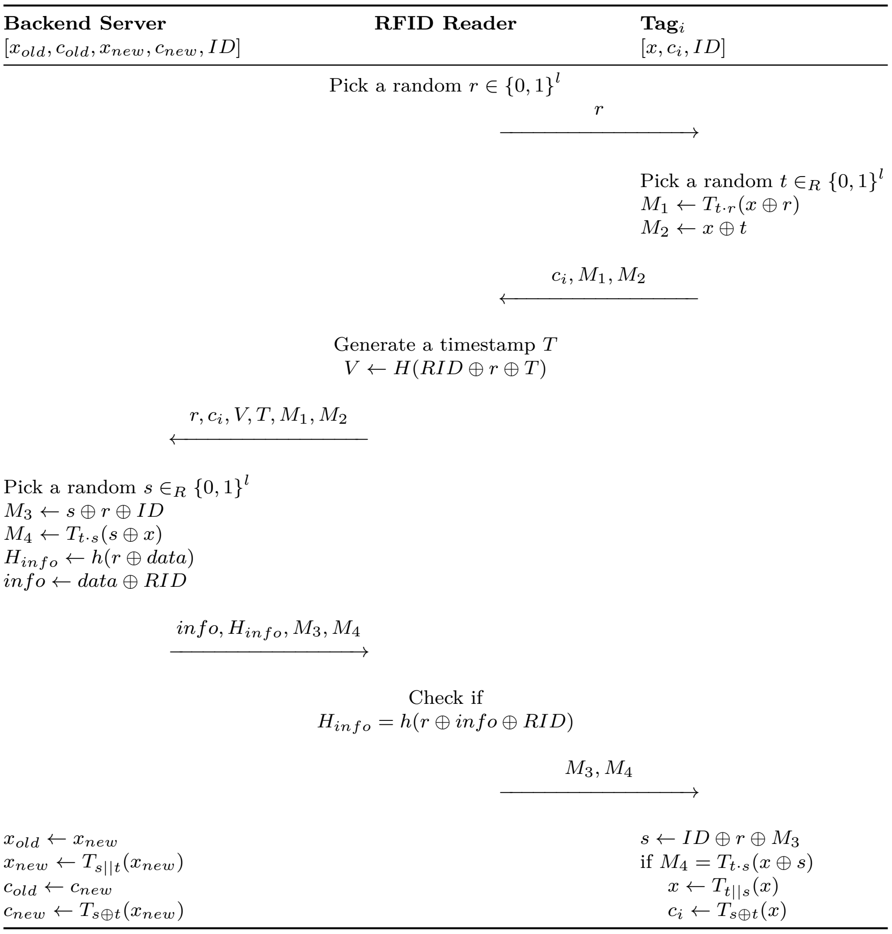
3.2.1 Protocol Description

Similar to Benssalah *et al.* , the scheme of Akgun *et al.*'s also consists of two stages: *initialization* and *authentication*. Figure 2 illustrates the authentication scheme of Akgun *et al.* .

The Initialization Stage: The server picks a random and different secret key for each tag. Then, the server saves the credentials of each tag in its own database, which has the form of $[x_{old}, c_{old}, x_{new}, c_{new}, ID]$ where c_{old}/c_{new} represents database index and ID is the identifier of a tag. Each tag stores $[x, c_i, ID]$ values in its non-volatile memory. RID denotes the identifier of a reader. The initial values of the database indexes and session secrets are set as $c_{new} \leftarrow c_{old} \leftarrow 0$ and $x_{new} \leftarrow x_{old} \leftarrow x$.

The Authentication Stage:

1. First of all, a legitimate RFID reader picks a random number $r \epsilon_R \{0,1\}^{\ell}$ and broadcasts it.
2. Once received r, the tag picks another random number $t \epsilon_R \{0,1\}^{\ell}$ and calculates $M_1 \leftarrow T_{t \cdot r}(x \oplus r)$ and $M_2 \leftarrow x \oplus t$ and forwards the triple messages (c_i, M_1, M_2) to the reader.
3. Next, the reader gets a time-stamp T, calculates $V \leftarrow H(RID \oplus r \oplus T)$ and sends back (r, c_i, V, T, M_1, M_2) to the server.
4. Upon receiving (r, c_i, V, T, M_1, M_2), the validity of V is first done. If V is valid, the server executes the subsequent transactions:
   (a) If c_i is equal to 0:
      - The server scans whole database and searches a match record. For each entry, the server computes $T_{old} \leftarrow T_{r \cdot (M_2 \oplus x_{old})}(x_{old} \oplus r)$ and $T_{new} \leftarrow T_{r \cdot (M_2 \oplus x_{old})}(x_{new} \oplus r)$. If M_1 is equal to T_{old} or T_{new}, the server finds the correct record in the database and then, the server sets $x \leftarrow x_{old}$ or $x \leftarrow x_{new}$ according to value of T_{old} or T_{new}.



Backend Server $[x_{old}, c_{old}, x_{new}, c_{new}, ID]$

RFID Reader

Tag$_i$ $[x, c_i, ID]$

Pick a random $r \in \{0,1\}^l$

$\xrightarrow{r}$

Pick a random $t \in_R \{0,1\}^l$
$M_1 \leftarrow T_{t \cdot r}(x \oplus r)$
$M_2 \leftarrow x \oplus t$

$\xleftarrow{c_i, M_1, M_2}$

Generate a timestamp T
$V \leftarrow H(RID \oplus r \oplus T)$

$\xleftarrow{r, c_i, V, T, M_1, M_2}$

Pick a random $s \in_R \{0,1\}^l$
$M_3 \leftarrow s \oplus r \oplus ID$
$M_4 \leftarrow T_{t \cdot s}(s \oplus x)$
$H_{info} \leftarrow h(r \oplus data)$
$info \leftarrow data \oplus RID$

$\xrightarrow{info, H_{info}, M_3, M_4}$

Check if
$H_{info} = h(r \oplus info \oplus RID)$

$\xrightarrow{M_3, M_4}$

$x_{old} \leftarrow x_{new}$
$x_{new} \leftarrow T_{s||t}(x_{new})$
$c_{old} \leftarrow c_{new}$
$c_{new} \leftarrow T_{s \oplus t}(x_{new})$

$s \leftarrow ID \oplus r \oplus M_3$
if $M_4 = T_{t \cdot s}(x \oplus s)$
$x \leftarrow T_{t||s}(x)$
$c_i \leftarrow T_{s \oplus t}(x)$

Fig. 2 The protocol of Akgun *et al.*

(b) If c_i is not equal to 0,
   - c_i is the current database index. The server gets the database record where c_i matches with c_{old} or c_{new} and executes $x \leftarrow x_{old}$ or $x \leftarrow x_{new}$.
   - The server checks the validity of M_1. If it is invalid, the tag would be rejected.

(c) The server picks another random number $s \epsilon_R \{0,1\}^{\ell}$ and computes $info \leftarrow RID \oplus data$, $H_{info} \leftarrow H(data \oplus r)$, $M_3 \leftarrow s \oplus r \oplus ID$, $M_4 \leftarrow T_{t \cdot s}(x \oplus s)$. Then, the server sends the quadruple $(info, H_{info}, M_3, M_4)$ to the reader.

(d) Next, the server executes $c_{old} \leftarrow c_{new}$, $x_{old} \leftarrow x_{new}$, $c_{new} \leftarrow T_{s \oplus t}(x_{new})$ and $x_{new} \leftarrow T_{t||s}(x_{new})$ for index and key updating.

5. The reader verifies the validity of H_{info}. If H_{info} is valid, the reader sends (M_3, M_4) to the tag.

6. The tag recovers s from M_3 by executing $s \leftarrow ID \oplus r \oplus M_3$.
7. The tag verifies checks the validity of M_4. If M_4 is valid, the tag executes $x \leftarrow T_{t||s}(x)$ and $c_i \leftarrow T_{s \oplus t}(x)$ to update the key and the index.

3.2.2 The security analysis of the protocol

In this section, we provide two impersonation attacks and a desyncronization attack that can be practically mounted to the protocol of Akgun *et al.* . In the first impersonation attack, the adversary actively monitors and modifies the messages between a victim tag and a legal reader, and the messages between the reader and the server. In the second impersonation attack, no tag is needed in order to impersonate the tag whose identity is stored in the first entry of the database. The desyncronization attack can applied to this protocol using the first impersonation attack.

Impersonation Tag-I: Our generic attack can be applied to this protocol in order to impersonate a victim tag $\mathcal{T}_0$ with another legal tag T_i. In this attack, the reader $\mathcal{R}$ authenticates $\mathcal{T}_j$ but the tag in front of $\mathcal{R}$ would be $\mathcal{T}_i$. The attack is performed as follows.

- $\mathcal{R}$ generates a random number r and sends it to the tag $\mathcal{T}_i$ but during the transmission $\mathcal{A}$ corrupts the message and set $r \leftarrow 0$.
- $\mathcal{T}_i$ also generates another random number t and computes M_1 and M_2. Note that since $r = 0$, $M_2 = T_{0.t}(x \oplus t) = 1$. $\mathcal{T}_i$ sends the triple (M_1, M_2, c_i) to $\mathcal{R}$.
- After the receiving the triple, $\mathcal{R}$ generates a timestamp T and computes $V \leftarrow H(RID \oplus r \oplus T)$ and sends the triple along with r, T and V to the server.
- During the transmission $\mathcal{A}$ sets $c \leftarrow 0$, $T \leftarrow T \oplus r$ and $r \leftarrow 0$.
- The server first validates the correctness of V by comparing it to $H(RID \oplus r \oplus T)$. This validation is passed because $RID \oplus 0 \oplus T \oplus r = RID \oplus T \oplus r$. Since $c_i = 0$, the server will perform a full search on the database. For the first element in the database, the server computes $T_{old} \leftarrow T_{(M_2 \oplus x_{old}).r}(x_{old})$ and $T_{new} \leftarrow T_{(M_2 \oplus x_{new}).r}(x_{new})$. Note that both T_{old} and T_{new} will be equal to 1 because of $r = 0$ and the property of Chebyshev polynomial $T_0(x) = 1$. Now, the server would see that T_{old} is equal to M_1. Then, the server computes M_3 and M_4 and sends them to $\mathcal{R}$. Finally, the server updates x_{old}, x_{new}, c_{old} and c_{new}.

After this attack, the reader authenticates the first entry in the database but the reader interacts with a different tag $\mathcal{T}_i$.

Impersonation Tag-II: In this attack, there is no need to interact with any tag in order to impersonate the first tag in the database of the server. The attack works as follows.

1. The reader sends a random r to $\mathcal{A}$ who try to impersonate $\mathcal{T}_0$.

2. $\mathcal{A}$ generates two random numbers t and x. Then, it sets $c_i \leftarrow 0$, $M_1 \leftarrow 1$ and $M_2 \leftarrow x \oplus t$ and sends the triple (M_1, M_2, c_i) to the reader.
3. Upon receiving the triple, $\mathcal{R}$ generates a timestamp T and computes $V \leftarrow H(RID \oplus r \oplus T)$. Next, it sends the triple along with r, T and V to the server.
4. During the transmission between $\mathcal{R}$ and the server, $\mathcal{A}$ sets $c \leftarrow 0$, $T \leftarrow T \oplus r$ and $r \leftarrow 0$.
5. The server first validates the correctness of V by comparing it to $H(RID \oplus r \oplus T)$. This validation is passed because $RID \oplus 0 \oplus T \oplus r = RID \oplus T \oplus r$. Since $c_i = 0$, the server will perform a full search on the database. For the first element of database, the server computes $T_{old} \leftarrow T_{(M_2 \oplus x_{old}).r}(x_{old})$ and $T_{new} \leftarrow T_{(M_2 \oplus x_{new}).r}(x_{new})$. Note that both T_{old} and T_{new} will be equal to 1 since $r = 0$ and due to the property of Chebyshev polynomials $T_0(x) = 1$. Now, the server would see that T_{old} is equal to M_1. Then, the server computes M_3 and M_4 and sends them to $\mathcal{R}$. Finally, the server updates x_{old}, x_{new}, c_{old} and c_{new}.

Desyncronization attack: The protocol is vulnerable to desynchronization attack. When two subsequent impersonation attacks are mounted, x_{old} and x_{new} are both updated incorrectly. This results in that the first tag in the database can no longer be authenticated by a legal reader.

4 Our Enhanced Protocol

In this section, we suggest another enhanced chaotic map-based RFID authentication protocol which conforms to EPCGlobal G2 C1 standard. Our protocol utilizes pseudo-random number generation, xor operation and algebraic properties of Chebyshev polynomials.

4.1 Protocol description

Figure 3 depicts our enhanced authentication protocol. The protocol consists of two phases: *initialization* and *authentication.*

4.1.1 The Initialization step

For each tag, the server generates a random secret key x. Next, the server saves the credentials of each tag in its own database that has the form of $[x_{old}, c_{old}, x_{new}, c_{new}, ID]$ where c_{old}/c_{new} represent database index values and ID is the identifier of a tag. Each tag stores $[x, c_i, ID]$ values in its non-volatile memory. RID denotes identifier of a reader. The initial values are set as $c_{new} \leftarrow c_{old} \leftarrow 0$ and $x_{new} \leftarrow x_{old} \leftarrow x$.

Backend Server	**RFID Reader**	**Tag**$_i$
$[x_{old}, c_{old}, x_{new}, c_{new}, ID]$		$[x, c_i, ID]$
	Pick a random $r \in_R \{0,1\}^\ell$	
	$\xrightarrow{r}$	
		Pick a random $t \in_R \{0,1\}^\ell$ $M_1 \leftarrow T_{x.t}(x \oplus r)$ $M_2 \leftarrow x \oplus t$
	$\xleftarrow{c_i, M_1, M_2}$	
	Generate a time-stamp T $V \leftarrow H(r, T, RID)$	
$\xleftarrow{c_i, r, V, T, M_1, M_2}$		
Pick a random $s \in_R \{0,1\}^\ell$ $M_3 \leftarrow T_{ID}(r \oplus t) \oplus s$ $M_4 \leftarrow T_{t \cdot s}(x \oplus s)$ $H_{info} \leftarrow H(r \oplus data)$ $info \leftarrow data \oplus RID$		
$\xrightarrow{info, H_{info}, M_3, M_4}$		
	Check if $H_{info} = H(info \oplus r \oplus RID)$	
	$\xrightarrow{M_3, M_4}$	
$x_{old} \leftarrow x_{new}$ $x_{new} \leftarrow T_{s\|\|t}(x_{new})$ $c_{old} \leftarrow c_{new}$ $c_{new} \leftarrow T_{s \oplus t}(x_{new})$		$s \leftarrow M_3 \oplus T_{ID}(r \oplus t)$ if $M_4 = T_{t \cdot s}(x \oplus s)$ $x \leftarrow T_{s\|\|t}(x)$ $c_i \leftarrow T_{s \oplus t}(x)$

Fig. 3 Our proposed RFID authentication protocol.

4.1.2 The Authentication step

The authentication of a tag is carried out by performing the following steps;

1. A legitimate RFID reader picks a random number $r \epsilon_R \{0,1\}^\ell$ and broadcasts it.
2. Once received r, the tag picks another random number $t \epsilon_R \{0,1\}^\ell$ and sets $M_1 \leftarrow T_{x.t}(x \oplus r)$ and $M_2 \leftarrow x \oplus t$. Next it sends the triple (M_1, M_2, c_i) to the reader.

3. After receiving (M_1, M_2, c_i), the reader generates a time-stamp T and computes $V \leftarrow H(RID, r, T)$. Next, it sends (M_1, M_2, c_i, r, V, T) to the server.
4. The server performs validity check on V. If it is valid, the server executes the following transactions:
   (a) If c_i is equal to 0:
      – The server scans the whole database and searches for a match. For each record, the server computes $T_{old} \leftarrow T_{(x_{old} \cdot (M_2 \oplus x_{old})}(x_{old} \oplus r)$ and $T_{new} \leftarrow T_{(x_{new} \cdot (M_2 \oplus x_{new}))}(x_{new} \oplus r)$. If M_1 is equal to T_{old} or T_{new}, the server finds the corresponding records in the database. The server sets either $x \leftarrow x_{old}$ or $x \leftarrow x_{new}$ according to the value of M_1.
   (b) If c_i is not equal to 0:
      – c_i is the database index of the current tag. The server gets the database record where c_i matches with c_{old} or c_{new} and updates $x \leftarrow x_{old}$ or $x \leftarrow x_{new}$.
      – The server checks the validity of M_1. If it is invalid, the server rejects the tag.
   (c) The server picks another random number $s \epsilon_R \{0,1\}^{\ell}$ and computes $info \leftarrow RID \oplus data$, $H_{info} \leftarrow H(r \oplus data)$, $M_3 \leftarrow T_{ID}(r \oplus t) \oplus s$ and $M_4 \leftarrow T_{t \cdot s}(x \oplus s)$. Next, it sends $(info, H_{info}, M_3, M_4)$ to the reader.
   (d) The server finally executes $x_{old} \leftarrow x_{new}$, $c_{old} \leftarrow c_{new}$, $x_{new} \leftarrow T_{t||s}(x_{new})$, and $c_{new} \leftarrow T_{s \oplus t}(x_{new})$ for key updating.
5. The reader verifies that H_{info} is valid. If H_{info} is valid, it sends the tuple (M_3, M_4) to the tag.
6. The tag recovers s from M_3 by computing $s \leftarrow M_3 \oplus T_{ID}(r \oplus t)$.
7. The tag verifies the validity of M_4. If M_4 is valid, the tag executes $c_i \leftarrow T_{s \oplus t}(x)$ and $x \leftarrow T_{s||t}(x)$ to update the index and the key, respectively.

4.2 Security Analysis of the Protocol

In this section, according to the security definitions described in Section 2.2, we provide the security analysis of our enhanced protocol. We prove our proposed protocol satisfies the security and privacy expectations.

Lemma 1 *Let $\mathcal{T}$ denote a victim legitimate tag in the RFID system. Then, without corrupting $\mathcal{T}$, the secret values of $\mathcal{T}$ cannot be computationally recovered.*

Proof For each query, $\mathcal{T}$ answer back with c_i, $M_1 = T_{x \cdot t}(x \oplus r)$ and $M_2 = x \oplus t$ values. Note that the value of t is randomly refreshed and this is directly used in the computation of M_1 and M_2. The computational adversary $\mathcal{A}$ cannot recover the values of t from M_1 and M_2 with non-negligible probability because of the fact that it is safeguarded by Chebyshev chaotic map hard problems (Definition 5 and Definition 6). On the other hand, the reader sends responses to the tag with $M_3 = T_{ID}(r \oplus t) \oplus s$ and $M_4 = T_{s \cdot t}(x \oplus s)$. M_3 and M_4 messages are randomized by s, which is randomly generated by the server and this value is unknown to $\mathcal{A}$. Similarly, A cannot recover s from M_3 and M_4 since the Chebyshev chaotic map hard problems (Definition 5 and Definition 6) safeguard s. Moreover, c_i is computed for database index and constructed by the previous x, s and t values using Chebyshev chaotic map hard problems. $\mathcal{A}$ still cannot retrieve x, s or t from c_i because of the Chebyshev chaotic map hardness property. Besides x and c_i values are updated after each successful session. Hence the old and new values of the x and c_i cannot be correlated with non-negligible probability.

Theorem 1 *Let ℓ denotes the security parameter of the RFID system. Then, our protocol achieves secure tag authentication given that $T.(.)$ is defined as a Chebyshev chaotic map with Definition 5 and Definition 6.*

Proof We assume that there is a polynomial-time bounded adversary $\mathcal{A}$ that impersonates a tag $\mathcal{T}_i$ to cheat a legal reader $\mathcal{R}$ with a non-negligible probability. Having received a random nonce r, $\mathcal{A}$ should compute $M_1 = T_{x^j \oplus t}(x^j \oplus r)$, $M_2 = x^j \oplus t$, and $c_i = c_i$. $\mathcal{A}$ has a chance to use the previously recorded responses of $\mathcal{T}_i$. For instance, during the protocol session π_j, $\mathcal{A}$ records the messages (r^j,M_1^j, M_2^j, M_3^j, M_4^j) between $\mathcal{R}$ and $\mathcal{T}_i$. $\mathcal{A}$ starts a new protocol session π_{j+1} with $\mathcal{R}$. $\mathcal{R}$ first sends r^{j+1} to $\mathcal{A}$. $\mathcal{A}$ chooses a new random t^{j+1} and has to compute the responses $M_1^{j+1} = T_{x^{j+1} \oplus t^{j+1}}(x^{j+1} \oplus r^{j+1})$ and $M_2^{j+1} = x^{j+1} \oplus t^{j+1}$. $\mathcal{A}$ needs the value of x^{j+1} in order to compute M_1^{j+1} and M_2^{j+1}. However, x^{j+1} can be computed by using t^j, s^j and x^j values but these values are also protected by the Chebyshev chaotic map because of Definition 5 and Definition 6. Therefore, $\mathcal{A}$ could use the previous response messages with a negligible probability of $2^{1-\ell}$.

Theorem 2 *Let ℓ denote the security parameter of the RFID system. Then, our protocol achieves secure reader authentication given that $T.(.)$ is defined as a Chebyshev chaotic map with Definition 5 and Definition 6.*

Proof We assume that there is a polynomial-time bounded adversary $\mathcal{A}$ that impersonates a legal reader $\mathcal{R}$ to cheat a tag $\mathcal{T}_i$ with a non-negligible probability. Having received M_1, M_2 and c_i, $\mathcal{A}$ needs to compute $M_3 = T_{ID}(r \oplus t) \oplus s$ and $M_4 \leftarrow T_{s \cdot t}(x \oplus s)$. $\mathcal{A}$ has a chance to use previously recorded responses of $\mathcal{R}$. For instance, during the protocol session π_j, $\mathcal{A}$ records the messages (r^j,M_1^j, M_2^j, M_3^j, M_4^j) between $\mathcal{R}$ and $\mathcal{T}_i$ but prevent $\mathcal{T}_i$ to receive M_3^j and M_4^j. Hence $\mathcal{T}_i$ could not update the values of x^j and c_i^j but the server did. Now, $\mathcal{A}$ starts a new protocol session π_{j+1} with $\mathcal{T}_i$. $\mathcal{A}$ first sends r^j to $\mathcal{T}_i$ and the tag chooses a new random t^{j+1} and sends back the responses $M_1^{j+1} = T_{x^j \oplus t^{j+1}}(x^j \oplus r^j)$ and $M_2^{j+1} = x^j \oplus t^{j+1}$. $\mathcal{A}$ needs the values of t^{j+1} and x^j in order to compute M_3^{j+1} and M_4^{j+1}. However, t^{j+1} and x^j are protected by Chebyshev chaotic map because of Definition 5 and Definition 6. So $\mathcal{A}$ could use the previous response messages with negligible probability of $2^{1-\ell}$.

Theorem 3 *Our proposed protocol achieves universal untraceability given that $T.(.)$ is an enhanced Chebyshev polynomial with Definition 5 and Definition 6.*

Proof Let $\mathcal{A}$ denote the adversary who achieves the experiment of universal untraceability with non-negligible probability. Let $\mathcal{T}_0$ and $\mathcal{T}_1$ be the victim tags and $\mathcal{R}$ denote the legitimate reader.

- Throughout the learning phase, $\mathcal{A}$ executes, eavesdrops, and drops the authentication sessions between $\mathcal{T}_0$ and $\mathcal{R}$ and the sessions between $\mathcal{T}_1$ and $\mathcal{R}$.
- During the challenge phase, let $\mathcal{R}$ perform successful authentications with $\mathcal{T}_0$ and $\mathcal{T}_1$, so both $\mathcal{T}_0$ and $\mathcal{T}_1$ updates their own secrets. After that, $\mathcal{A}$ executes, eavesdrops, and breaks authentication sessions between $\mathcal{T}_b$ and $\mathcal{R}$ where b is chosen randomly.
- In the guess phase, $\mathcal{A}$ has to guess the value of b. $\mathcal{A}$ tries to correlate the responses of the tags in the learning phase and the responses of the $\mathcal{T}_b$ in the challenge phase. Since the secret values of $\mathcal{T}_b$ are updated using the help of the Chebyshev chaotic map with Definition 5 and Definition 6, at least one successful authentication yields the responses to be indistinguishable. Therefore, $\mathcal{A}$ cannot correlate these responses.

Theorem 4 *Our enhanced protocol does not achieve existential untraceability.*

Proof Let $\mathcal{A}$ be an adversary that executes two different successful authentication runs (say π_1 and π_2) with a victim tag $\mathcal{T}_0$. It is clearly seen that if a legitimate reader does not execute any successful authentication with $\mathcal{T}_0$ between π_1 and π_2, the database index c_i in these runs will be the same. This information gives $\mathcal{A}$ to distinguish these two runs. Hence, our protocol does not satisfy existential untraceability.

Theorem 5 *Our protocol satisfies forward privacy given that $T.(.)$ is an enhanced Chebyshev polynomial with Definition 5 and Definition 6.*

Proof Let $\mathcal{A}$ denote the adversary who achieves the experiment of forward privacy with a non-negligible probability. Let $\mathcal{T}_0$ and $\mathcal{T}_1$ be the victim tags and $\mathcal{R}$ denote the legitimate reader.

- Throughout the learning phase, $\mathcal{A}$ executes, eavesdrops, and drops the authentication sessions between $\mathcal{T}_0$ and $\mathcal{R}$ and the sessions between $\mathcal{T}_1$ and $\mathcal{R}$.
- During the challenge phase, let $\mathcal{R}$ perform successful authentications with $\mathcal{T}_0$ and $\mathcal{T}_1$, so both $\mathcal{T}_0$ and $\mathcal{T}_1$ updates their own secrets. After that, $\mathcal{A}$ executes, eavesdrops, and breaks authentication sessions between $\mathcal{T}_b$ and $\mathcal{R}$ where b is chosen randomly. Also, The internal state of the tag $\mathcal{T}_b$ is given to $\mathcal{A}$. $\mathcal{A}$ learns ID, x^{j+1}, c_b^{j+1} such that $x^{j+1} = T_{s^j||t^j}(x^j)$, $c_b^{j+1} = T_{s^j \oplus t^j}(x^j)$.
- In the guess phase, $\mathcal{A}$ has to predict the correct value of b. $\mathcal{A}$ tries to find out the correlation between the authentication messages eavesdropped in the learning phase and the current internal state of $\mathcal{T}_b$. Assume that $\mathcal{A}$ recorded an authentication run that occurred during the challenge phase. After the challenge phase, the secret key and index values of $\mathcal{T}_b$ are updated.

The recorded authentication messages are as follows:

$$\begin{aligned} M_1^j &= T_{x^j \cdot t^j}(x^j \oplus r^j) \qquad (1)\\ M_2^j &= x^j \oplus t^j \\ M_3^j &= T_{ID}(r^j \oplus t^j) \oplus s^j \\ M_4^j &= T_{s^j \cdot t^j}(x^j \oplus s^j) \\ c_i^j & \\ r^j & \end{aligned}$$

Since ID is fixed through the life of $\mathcal{T}_i$, M_3^j would better for $\mathcal{A}$ to find a correlation. $\mathcal{A}$ has to know t^j and s values. In order to recover t^j value, $\mathcal{A}$ should know the secret x^j. Using x^{j+1} which is given to $\mathcal{A}$ in the learning phase, it is not computationally feasible to extract x^j because of the property of Chebyshev chaotic map. Moreover, s value cannot be retrieved from M_4^j due to hardness of Chebyshev chaotic map problems (see Definitions 5 and 6). As a result, $\mathcal{A}$ achieves forward privacy experiment with a negligible probability.

4.3 Performance Comparisons

Table 2 Security and privacy comparison of the protocols.

Security features	[35]	[36]	[11]	[6]	[3]	Improved Protocol
Mutual authentication	✔	✘	✘	✘	✘	✔
Mobility	✔	✔	✘	✔	✔	✔
Resistance to server impersonation attack	✔	✔	✘	✘	✔	✔
Resistance to tag impersonation attack	✔	✔	✘	✘	✘	✔
Resistance to secret disclosure attacks	✔	✘	✘	✘	✔	✔
Resistance to replay attacks	✔	✔	✘	✘	✔	✔
Resistance to de-synchronization attacks	✔	✘	✘	✘	✘	✔
Backward untraceability	✘	✘	✘	✘	✔	✔
Universal untraceability	✘	✘	✘	✘	✔	✔

Table 3 Performance comparison of the protocols.

Features	[35]	[36]	[11]	[6]	[3]	Improved Protocol
Communication rounds	5	5	5	5	5	5
Communication cost	4M	4M	3M	4M	3M	3M
Storage overhead	4M	4M	3M	4M	3M	3M
Back-end server computation	3SRS	1R+4P	1R+2T	1R+4T	1R+4T+2H	1R+5T+2H
Reader computation	1R	1R+1H	1R	1R+2H	1R+2H	1R+2H
Tag computation	3S+4H+1R	1R+6P	1R+4T	1R+4T	1R+4T	1R+5T
Key search complexity	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
Crypto primitives	X,P	X,P	T,X,C	T,X,C	T,X,C	T,X,C

RFID tags are resource constrained devices, *i.e.*, they have low computation powers and small storage capacities. Consequently, RFID protocols employ resource friendly cryptographic mechanisms. In our protocol, tags generate pseudo random numbers, perform xor operations, concatenate bit strings and operate on Chebyshev chaotic maps.

Chebyshev polynomials are employed in various cryptographic schemes including key agreement protocols [26,34], password-based authentication schemes [16,20], public key encryption schemes [37,8], and RFID authentication protocols [11,6,3]. The implementation of Chebyshev polynomials uses small resources so that smart cards and RFID tags can utilize them.

From performance point of view, the most critical operation in our scheme is the semi-group property of enhanced Chebyshev chaotic maps, since other operations performed by tag are insignificant such as xor operation and concatenation of bit strings. The number of steps required for the computation of $T_n(x)$ grows linearly with n which is later reduced to logarithmic complexity in [17] by making the following observation:

$$T_{2n}(x) = T_2(T_n(x)) \tag{2}$$

$$T_{2n+1}(x) = 2.T_{n+1}(x).T_n(x) - x \tag{3}$$

A number of methods have been proposed to reduce the required computation time of $T_n(x)$ in [32]. Moreover, the trigonometric equality $T_n(x) = cos(n.arccos(x))$ allows us to compute Chebyshev polynomials more efficiently and implement them on resource constrained devices such as RFID tags.

Security characteristics of our improved protocol is compared with the existing protocols and the findings are summarized in Table 2. Our improved protocol supports all the desired features which are expected from an RFID authentication protocol. We would provide existential untraceability feature in our protocol by avoiding the c_i variable however, this would cause each authentication request to require a search operation with linear complexity.

Let C denote concatenation, H denote hash function, M denote message length, P denote pseudo-random number generation, R denote random number generation, S denote modular squaring, SRS denote square root solving, T denote Chebyshev polynomials and X denote xor operation. We compare the computational complexity of our protocol with previous works in Table 3. The result shows that, our protocol does not exceed the required computation power of previous works though it provides a better security.

5 Conclusion

In this paper, we investigated the security and privacy of two recently published RFID authentication protocols. The former protocol employs Chebyshev polynomials and we proved that this protocol does not provide resistance against tag impersonation, tracking, and de-synchronization attacks. On the other hand, the latter protocol is the enhanced version of the first one. Notwithstanding the proposed improvements, the latter protocol also contains basic security weaknesses and it is not resistant against tag impersonation, and de-synchronization attacks. Moreover, we offered an improved RFID authentication protocol that employs chaotic maps for low-cost devices. Our protocol complies with the EPC C1-G2 standard and achieves the expected security and privacy requirements.

References

1. EPCglobal, EPC radio-frequency identity protocols class 1 generation 2 UHF RFID protocol for communications at 860MHz–960 MHz, Version 1.2.0, Specification for RFID Air Interface (2008)
2. Akgun, M., Bayrak, A.O., Caglayan, M.U.: Attacks and improvements to chaotic map-based rfid authentication protocol. Security and Communication Networks **8**(18), 4028–4040 (2015). DOI 10.1002/sec.1319
3. Akgun, M., Uekae, T., Caglayan, M.: Vulnerabilities of rfid security protocol based on chaotic maps. In: Network Protocols (ICNP), 2014 IEEE 22nd International Conference on, pp. 648–653 (2014). DOI 10.1109/ICNP.2014.103
4. Alomair, B., Clark, A., Cuellar, J., Poovendran, R.: Scalable rfid systems: a privacy-preserving protocol with constant-time identification. Dependable Systems and Networks, International Conference on **0**, 1–10 (2010)
5. Avoine, G.: Adversarial model for radio frequency identification. Cryptology ePrint Archive, Report 2005/049 (2005). http://eprint.iacr.org/
6. Benssalah, M., Djeddou, M., Drouiche, K.: Security enhancement of the authenticated rfid security mechanism based on chaotic maps. Security and Communication Networks **7**(12), 2356–2372 (2014). DOI 10.1002/sec.946
7. Berbain, C., Billet, O., Etrog, J., Gilbert, H.: An efficient forward private rfid protocol. In: Proceedings of the 16th ACM Conference on Computer and Communications Security, CCS '09, pp. 43–53. ACM, New York, NY, USA (2009). DOI 10.1145/1653662.1653669
8. Bergamo, P., D'Arco, P., De Santis, A., Kocarev, L.: Security of public-key cryptosystems based on chebyshev polynomials. IEEE Transactions on Circuits and Systems I: Regular Papers **52**(7), 1382–1393 (2005)
9. Burmester, M., de Medeiros, B., Motta, R.: Anonymous rfid authentication supporting constant-cost key-lookup against active adversaries. IJACT **1**(2), 79–90 (2008)
10. Chen, Y., Chou, J.S., Sun, H.M.: A novel mutual authentication scheme based on quadratic residues for {RFID} systems. Computer Networks **52**(12), 2373 – 2380 (2008)

11. Cheng, Z.Y., Liu, Y., Chang, C.C., Chang, S.C.: Authenticated rfid security mechanism based on chaotic maps. Security and Communication Networks **6**(2), 247–256 (2013). DOI 10.1002/sec.709
12. Chien, H.Y.: Sasi: A new ultralightweight rfid authentication protocol providing strong authentication and strong integrity. Dependable and Secure Computing, IEEE Transactions on **4**(4), 337–340 (2007)
13. Chien, H.Y., Chen, C.H.: Mutual authentication protocol for rfid conforming to epc class 1 generation 2 standards. Comput. Stand. Interfaces **29**(2), 254–259 (2007). DOI 10.1016/j.csi.2006.04.004
14. Chien, H.Y., Huang, C.W.: A Lightweight Authentication Protocol for Low-Cost RFID. Journal of Signal Processing Systems **59**, 95–102 (2010). DOI 10.1007/s11265-008-0281-8
15. Coisel, I., Martin, T.: Untangling RFID privacy models. Journal of Computer Networks and Communications (2013)
16. Farash, M.S., Attari, M.A.: An efficient and provably secure three-party password-based authenticated key exchange protocol based on chebyshev chaotic maps. Nonlinear Dynamics **77**(1-2), 399–411 (2014)
17. Fateman, R.J.: Lookup tables, recurrences and complexity. In: Proceedings of the ACM-SIGSAM 1989 International Symposium on Symbolic and Algebraic Computation, ISSAC '89, pp. 68–73 (1989)
18. Fernandez-Mir, A., Trujillo-Rasua, R., Castella-Roca, J.: Scalable RFID Authentication Protocol Supporting Ownership Transfer and Controlled Delegation. In: Workshop on RFID Security – RFIDSec'11. Amherst, Massachusetts, USA (2011)
19. Garfinkel, S., Rosenberg Beth, .: RFID : applications, security, and privacy. Boston, Mass. ; London : Addison-Wesley (2005). Formerly CIP
20. Guo, C., Chang, C.C.: Chaotic maps-based password-authenticated key agreement using smart cards. Communications in Nonlinear Science and Numerical Simulation **18**(6), 1433–1440 (2013)
21. Ha, J., Moon, S.J., Nieto, J.M.G., Boyd, C.: Low-cost and strong-security rfid authentication protocol. In: EUC Workshops, pp. 795–807 (2007)
22. Juels, A.: Minimalist Cryptography for Low-Cost RFID Tags. In: C. Blundo, S. Cimato (eds.) International Conference on Security in Communication Networks – SCN 2004, *Lecture Notes in Computer Science*, vol. 3352, pp. 149–164. Springer, Amalfi, Italy (2004)
23. Liu, Y.: An efficient rfid authentication protocol for low–cost tags. In: Proceedings of the 2008 IEEE/IFIP International Conference on Embedded and Ubiquitous Computing – Volume 02, EUC'08, pp. 180–185. IEEE Computer Society, Washington, DC, USA (2008)
24. Lo, N., Yeh, K.H., Yeun, C.Y.: New mutual agreement protocol to secure mobile rfid-enabled devices. Information Security Technical Report **13**(3), 151 – 157 (2008)
25. Maimut, D., Ouafi, K.: Lightweight cryptography for rfid tags. IEEE Security & Privacy **10**(2), 76–79 (2012)
26. Niu, Y., Wang, X.: An anonymous key agreement protocol based on chaotic maps. Communications in Nonlinear Science and Numerical Simulation **16**(4), 1986 – 1992 (2011)
27. Ohkubo, M., Suzuki, K., Kinoshita, S.: Cryptographic Approach to 'Privacy-Friendly' Tags. In: RFID Privacy Workshop. MIT, Massachusetts, USA (2003)
28. Paise, R.I., Vaudenay, S.: Mutual authentication in rfid: Security and privacy. In: Proceedings of the 2008 ACM Symposium on Information, Computer and Communications Security, ASIACCS '08, pp. 292–299. ACM, New York, NY, USA (2008). DOI 10.1145/1368310.1368352. URL http://doi.acm.org/10.1145/1368310.1368352
29. Song, B., Mitchell, C.J.: Scalable RFID Security Protocols supporting Tag Ownership Transfer. Computer Communication, Elsevier (2010)
30. Tian-tian, Y., Quan-yuan, F.: A security rfid authentication protocol based on hash function. In: Information Engineering and Electronic Commerce, 2009. IEEC '09. International Symposium on, pp. 804–807 (2009). DOI 10.1109/IEEC.2009.174
31. Vaudenay, S.: On Privacy Models for RFID. In: K. Kurosawa (ed.) Advances in Cryptology – Asiacrypt 2007, *Lecture Notes in Computer Science*, vol. 4833, pp. 68–87. Springer, Kuching, Malaysia (2007)

32. Wang, X., Zhao, J.: An improved key agreement protocol based on chaos. Communications in Nonlinear Science and Numerical Simulation **15**(12), 4052 – 4057 (2010)
33. Wong, K.: A fast chaotic cryptographic scheme with dynamic look-up table. Physics Letters A **298**(4), 238–242 (2002)
34. Xing-Yuan, W., Da-Peng, L.: A secure key agreement protocol based on chaotic maps. Chinese Physics B **22**(11), 110,503 (2013)
35. Yeh, T.C., Wang, Y.J., Kuo, T.C., Wang, S.S.: Securing {RFID} systems conforming to {EPC} class 1 generation 2 standard. Expert Systems with Applications **37**(12), 7678 – 7683 (2010)
36. Yoon, E.J., Jeon, I.S.: An efficient and secure Diffie–Hellman key agreement protocol based on Chebyshev chaotic map. Communications in Nonlinear Science and Numerical Simulation **16**(6), 2383–2389 (2011)
37. Zhang, L.: Cryptanalysis of the public key encryption based on multiple chaotic systems. Chaos, Solitons & Fractals **37**(3), 669–674 (2008)