Enhanced Mutual Authentication Scheme based on Chaotic Map for PCM in NFC Service Environment

Sung-Wook Park and Im-Yeong Lee

Department of Computer Science Engineering, Soonchunhyang University
Asan, South Korea
[e-mail: swpark@sch.ac.kr, imylee@sch.ac.kr]
*Corresponding author: Im-Yeong, Lee

Received September 21, 2016; revised December 27, 2016 ; accepted February 15, 2017; published February 28, 2017

Abstract

Currently, automated payment services provide intuitive user interfaces by adapting various wireless communication devices with mobile services. For example, companies like Samsung, Google, and Apple have selected the NFC payment method to service payments of existing credit cards. An electronic payment standard has been released for NFC activation within Korea and will strengthen the safety of payment service communications. However, there are various security risks regarding the NFC-based electronic payment method. In particular, the NFC payment service using the recently released lightweight devices cannot provide the cryptographic strength that is supported by many financial transaction services. This is largely due to its computational complexity and large storage resource requirements. The chaotic map introduced in this study can generate a highly complicated code as it is sensitive to the initial conditions. As the lightweight study using the chaotic map has been actively carried out in recent years, associated authentication techniques of the lightweight environment have been released. If applied with a chaotic map, a high level of cryptographic strength can be achieved that can provide more functions than simple XOR operations or HASH functions. Further, this technique can be used by financial transaction services. This study proposes a mutual authentication technique for NFC-PCM to support an NFC payment service environment based on the chaotic map.

***Keywords*:** NFC Mobile, Chebyshev Polynomial, Mutaul Authentication, Light-weight

A preliminary version of this paper was presented at APIC-IST 2016, and was selected as an outstanding paper.
This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the MEST (Ministry of Education, Science and Technology) (2013R1A1A2012940) and the Soonchunhyang University.
This research was supported by the MSIP(Ministry of Science, ICT and Future Planning), Korea, under the ITRC(Information Technology Research Center) support program (IITP-2016-R0992-16-1006) supervised by the IITP (Institute for Information & communications Technology Promotion)

https://doi.org/10.3837/tiis.2017.02.032 **ISSN : 1976-7277**

1. Introduction

Using a mobile payment service stored on a smartphone is becoming very popular as the payment method of the future. In the past, monetary transactions were settled via exchange of seashells or metallic currency. In the 7th Century B.C., Persia produced the very first coin consisting of mixed gold and silver with a consistent size and weight. Paper money appeared in the Song Dynasty of the 11th century, which was used as written proof of deposit. In the 20th century, the era of credit card was initiated when the restaurant-only Diner's Club card appeared in New York in 1950. More recent automated payment services complement the weaknesses of the traditional automated payment methods and create an intuitive payment service by combining various wireless devices and mobile services. Google, Samsung, and Apple have initiated proprietary mobile payment services. These services are based on new payment business models that utilize the lightweight NFC-based devices and are entering the market with high frequency. Unlike the typical business models, such payment services must "add value" in the sense that they provide more function and are easier to use. In particular, the new models must properly protect the users' financial properties and personal private information. To satisfy this requirement, relevant fields within academia and research institutes announced standard techniques that will, for the current state of technology, provide a robust level of protection. However, the technology is in a state of constant change that is driven by the changing business models. In particular, as smart devices become more and more lightweight, it is difficult to implement the necessary security technology with "industrial cryptographic strength." It is easy to understand why this is the case. The current mobile payment technology possesses the computational power and storage assigned to the mobile device by the manufacturer. "Industrial cryptographic strength" security and cryptographic algorithms consume large amounts of processor cycles and have very large storage requirements. Many recently released NFC payment techniques use PCM (passive communication mode) of NFC, thus making it more difficult to apply cryptographic techniques that are sufficiently strong to satisfy the payment standard in terms of both storage requirements and computational power. This paper proposes a new lightweight mutual authentication system using a chaotic map. This paper is structured as follows: Chapter 2 introduces relevant technologies to help understand the proposed method, and Chapter 3 analyzes the security requirements required by the proposed method based on the existing studies. Chapter 4 documents the proposed system that satisfies the security requirements, and Chapter 5 analyzes the proposed scheme along with the security requirements. Finally, Chapter 6 provides concluding comments.

II. Related Researches

In this Section, we describe the NFC support mode and analyze a viable authentication scheme based on the chaotic map in the NFC passive communication mode.

2.1 NFC operation mode

The NFC operation mode can be largely divided into two modes: (1) ACM (active communication mode) and (2) PCM. The ACM communicates while the initiator device, the target device alternately generates the RF field, and the device waiting for the data deactivates the RF field. Each device has its own power supply. The PCM draws the operating power from the electromagnetic field generated by the initiator and identifies the target device as the responding device. The target device can operate in a semi-permanent manner without an additional power supply if the electromagnetic field from the initiator exists. In an NFC-based payment service environment, there are a number of PCM-based NFC payment services. However, such payment services cannot provide strong security support, which makes the device vulnerable to leakage of personal information. This study proposes a PCM-based user authentication scheme that considers an environment characterized by (1) limited computational capability and (2) limited communication when compared to the ACM.

2.2 Business Model

As electronic payment services using NFC have recently grown in popularity, various electronic payment business models have dominated the market. Samsung's GEAR S2 is a good example. GEAR S2 offers a contactless payment transaction service based on NFC technology. In particular, GEAR S2 safely stores and manages data including encryption key, fingerprint information, and the user's credential information using the ESE (embedded secure element). Furthermore, ESE supports EMVCo, which enables worldwide inter-operability and secured payment transactions. This product has acquired the certificate of CC EAL5+ (high). However, as this is a proprietary product, the detailed internal structure of the system is not known. When considering the problems in smart watches (battery, storage space, computational power), it seems that a high level of cryptographic strength cannot be provided. Other examples of similar business models are "LoopPay," a contactless payment transaction service developed by LoopPay, a US start-up, and the "COIN" solution, developed by OnlyCoin in San Francisco. Recently, to target the 2016 summer Olympics, VISA, a credit card company, has announced the "NFC Ring," a ring-shaped device that supports face-to-face transactions using NFC-supporting devices (**Fig. 1**). These services represent exemplars of the business model and the environment described in this paper.

Fig. 1. Example of Business Model (NFC Ring)

2.3 Chebyshev Chaotic Map

A chaotic map reacts sensitively to initial conditions and generates sequences of great random number characteristics and these have made the chaotic map useful as the basic element of various cryptographic algorithms and many new cryptographic technologies have been proposed [1-3,6-11]. If the accurate initial value and the parameter of the chaotic function are unknown, the output value is unpredictable and makes it seem random. Such properties correspond with the algorithmic characteristics required in the field of cryptography, and because most chaotic-map-based systems implementing cryptography require large computational resources during encryption and decryption, it is nearly impossible to deploy this technology within a lightweight environment. However, various authentication schemes that can be implemented in lightweight environments like RFID have been proposed using a chaotic map. This study proposes a new lightweight authentication scheme using the chaotic map. The study attempts to apply the chaotic technique for the following reasons:

- The sensitive property of a chaotic map to initial conditions can lead to a very complicated passcode.
- If applied within a lightweight environment, a high level of cryptographic strength can be implemented while use of simple XOR operations or HASH function cannot satisfy the requirements. Therefore, a chaotic technique can be used within the financial transaction service sector.

Definition 1

- In the case of $n > 1$, *n-th* order Chebyshev polynomial map $T_n : [-1, 1] \rightarrow [-1, 1]$ is the density does not change.
- Chebyshev Polynomial $T_n(x)$ is defined as follows, n is an integer, X is adjustable from $[-1, 1]$.

$$T_n(x) = \cos(n \arccos x) \ (-1 \leq x \leq 1) \tag{1}$$

When you organize an expression that is repeated, it is as follows :

$T_0(x)=1, \quad T_1(x)=x, \quad T_2(x)=2x^2-1, \ldots$ (2)
$T_{n+1}(x)=2xT_n(x) - T_{n-1}(x), \quad n=1,2,\ldots$

Definition 2
- When x and y are given, to find the n is the DLP(Discrete Logarithm Problem).

$T_n(x)=y$ (3)

Definition 3
- When x, $T_r(x)$ and $T_s(x)$ are given, to find a $T_{rs}(x)$ is a DHP (Diffie-Hellman Problem).

Definition 4
- Semi-group Property

$T_s(T_r(x)) = \cos(r \arccos(\cos(s \arccos x)))$
$= \cos(rs \arccos x)$ (4)
$= \cos(sr \arccos x)$
$= \cos(s \arccos(\cos(r \arccos x)))$
$= T_s(T_r(x))$

Definition 5
- *Enhanced Chebyshev Polynomials*

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

2.4 Based research of Chaotic maps

The RFID authentication study has progressed over time, and solutions for many authentication issues of the passive tag have been proposed. As the chaotic cryptographic technology has recently received attention, studies to apply this technology to RFID have been conducted. However, most chaotic cryptographic technology requires a high level of computational resources and it is difficult for a lightweight system to satisfy these resource requirements. However, in 2003, *Lee et al*. proved that the computational time complexity of the Chebyshev polynomial-based chaotic cryptographic algorithm, possesses similar performance as HASH, and the chaotic cryptographic scheme was brought up as a topic of discussion again [1]. In 2013, Cheng et al. proposed a RFID authentication protocol based on the Chebyshev Polynomial [5]. However, its vulnerability to de-synchronization attack and secret information exposure attack were revealed in 2014 by *Akgun et al*. *Akgun et al*. proposed a new protocol to solve the issue [6, 7]. However, the author himself/herself stated that this method provides only a temporary solution with little gains in efficiency. Later in 2014, *Benssalah et al*. used an improved Chebyshev Polynomial algorithm (Definition 5) and proposed another authentication protocol that demonstrated improvement relative to the scheme of *Cheng et al* [8]. The scheme proposed by *Benssalah et al*., however, contains logic errors in computing message length and this makes it possible for the attacker to generate a random, wanted message and authentication value using tracking attack and tag impersonation

attach [9]. Building on this result, *Akgun et al*. removed the M_1 section of the logical error and added additional values to M_2 as the authentication value to mitigate these vulnerabilities.

2.5 The Protocol of Cheng et al

Here we analyze the technique proposed by *Cheng et al*. We then explain the secret disclosure attack, one of the several known attack schemes. A more detailed protocol is described below.

2.5.1 Protocol Description

Step 1 : The Reader (R) generates a random number *r* and transmits it to the Tag (T).

$$R : r \in \{0,1\}^r$$
$$R \rightarrow T : r$$

Step 2 : The tag, after receiving r, generates a random number t and computes messages, M_1, M_2, and M_3. M_1 consists of tag *ID*, random number *t* and *r*, and these are combined by XOR ope ration. M_2 is chaotic encryption message encrypted by random number *r* and *t*. Last, M_3 consists of the session key *x* and random number *t* between the tag and the server, and these are the XOR operation with each other. Next, the tag transmits the generated each message to the Reader.

$$T : t \in \{0,1\}^t$$
$$T : M_1 = h(ID) \oplus t \oplus r$$
$$T : M_2 = (T_{r,t}(x)) \quad (6)$$
$$T : M_3 = x \oplus t$$
$$T \rightarrow R : M_1, M_2, M_3$$

Step 3 : The Reader, upon receiving the messages from the tag, transmits the random information *r* and M_1, M_2, and M_3 to the Server(S). The Server, after receiving the messages, generates a value $h(ID) \oplus x$ through XOR operation of M_1, M_3 and *r*, and checks the validity of message via database record search . Last, verifies the identity of the tag.

$$R \rightarrow S : r,\ M_1, M_2, M_3$$
$$S : h(ID) \oplus x = M_1 \oplus M_3 \oplus r \quad (7)$$
$$S : (h(ID) \oplus x)' =? h(ID) \oplus x$$

Step 4 : The Server (S) obtains the *t* via XOR operation using random number *r*, message M_1 and tag's *ID* obtained through database indexing, and checks the validity of M_2 via computed $T_r(T_t(x))$.

$$S : t = M_1 \oplus h(ID) \oplus r$$
$$S : M_2 =? T_r(T_t(x)) \quad (8)$$

Step 5 : If M_2 is valid, the server generates random information *s* and M_4. If not valid, the session is closed.

$$S : S \in \{0, 1\}^l \quad (9)$$
$$S : M_4 = H(ID) \oplus r \oplus s$$

Step 6.1 : At the verification stage, the value of M_2, received by the server is validated relative to the Tag. If M_2, equals $T_r(T_t(x))$, then x, x_{old} are computed respectively for replacement..

$$S : if\ M_2 = T_r(T_t(x))$$
$$S : M_5 = T_{s,t}(x) \quad (10)$$
$$S : x_{old} \leftarrow x$$
$$S : x \leftarrow x \oplus (t||s)$$

Step 6.2 : At the verification stage, if the value of M_2, received from the Tag, is equal to $T_r(T_t(x_{old}))$, then x, x_{old} are computed respectively for replacement. Later, the server transmits the generated messages M_4, and M_5 to the tag.

$$S : if\ M_2 = T_r(T_t(x_{old}))$$
$$S : M_5 = T_{s,\,t}(x_{old}) \quad (11)$$
$$S : x \leftarrow x_{old} \oplus (t||s)$$
$$S \rightarrow T : M_4, M_5$$

Step 7 : The tag, after receiving the messages from the server, computes s and $T_s(T_t(x))$ and checks the validity of M_5. The tag obtains the s via XOR operation using random number r, message M_4 and tag's $h(ID)$. After, the value of x is then updated to $x \oplus (t||s)$.

$$T : s = M_4 \oplus h(ID) \oplus r$$
$$T : if\ M_5 = T_{s,t}(x) \quad (12)$$
$$T : x \leftarrow x \oplus (t||s)$$

2.5.2 Attack

The main observation of the Cheng et al. method regards the shared secret key x for update. This protocol validates the message through Chebyshev cryptographic techniques followed by updating the secret key x; the secret information $(t||s)$ can be obtained from the messages exchanged over the public channel.

Can disclose the $(t||s)$ as follows:

The attacker taps the communication messages of the first session and acquires the messages M_1, M_2, and M_3.

$$M_1 = H(ID) \oplus t \oplus r$$
$$M_2 = (T_{r,t}(x)) \quad (13)$$
$$M_3 = x \oplus t$$

(1) Next, the attacker taps the communication messages of the second session and acquires the messages M_1', M_2', and M_3'.

$$M_1' = H(ID) \oplus t' \oplus r'$$
$$M_2' = (T_{r',t'}(x')) \quad (14)$$
$$M_3' = x' \oplus t'$$

(2) Then, the acquired messages are combined to derive the following values.

$$M_1 \oplus M_3 \oplus r = H(ID) \oplus t \oplus r \oplus x \oplus t \oplus r \quad (15)$$
$$= H(ID) \oplus x$$

$$M_1' \oplus M_3' \oplus r' = H(ID) \oplus t' \oplus r' \oplus x' \oplus t' \oplus r' \quad (16)$$
$$= H(ID) \oplus x'$$

(3) At the key update stage, x is updated as follows, and therefore, it is possible to find $(t||s)$.

$$M_1 \oplus M_3 \oplus r \oplus M_1' \oplus M_3' \oplus r' = h(ID) \oplus t \oplus r \oplus x \oplus t \oplus r \oplus H(ID) \oplus t' \oplus r' \oplus x' \oplus t' \oplus r' \quad (17)$$
$$= h(ID) \oplus x \oplus H(ID) \oplus x'$$
$$= x \oplus x' \ , \ x' \leftarrow x \oplus (t||s)$$
$$= x \oplus x \oplus (t||s)$$
$$= t||s$$

Like this, If t and s are exposed, the secret key x and $H(ID)$ are all exposed, and therefore, the attacker becomes capable of randomly generating the authentication information $T^*(x)$.

2.6 The Protocol of Benssalah et al

Here we analyze the technique proposed by *Benssalah et al*. We then explain the Tracking attack and Tag impersonation attack, the several known attack schemes. A more detailed protocol is described below.

2.6.1 Protocol Description

Step 1 : The reader (R) generates a random number r and transmits it to the tag (T).

$R : r \in \{0,1\}^r$

$R \rightarrow T : r$

Step 2 : The tag, after receiving r, generates a random number t and computes M_1, M_2, and M_3. M_1 consists of tag *ID*, *h(ID)*, random number t and r, and these are combined by XOR operation. M_2 is chaotic encryption message encrypted by random number r and t. Last, M_3 consists of the session key x and random number t between the tag and the server, and these are the XOR operation with each other. Next, the tag transmits the generated each message to the reader.

$T : t \in \{0,1\}^t$

$T : M_1 = h(ID) \oplus ((r \oplus t)||(t \oplus ID)) \oplus t$

$T : M_2 = T_{r,t}(x)$ (18)

$T : M_3 = x \oplus t$

$T \rightarrow R : M_1, M_2, M_3$

Step 3 : The reader, after receiving the messages from the Tag, generates the time stamp T and computes the following. Message V generated by reader is a authentication data for verifying the reader by server. V consists of *RID* of reader, random number r and time-stamp T which are performed by a hash function. Later, the server identifies the *ID* of the reader through to validation process using V, r and T. Next, the reader transmits M_1, M_2, M_3, C_i, r, V, and T to the server.

$$R : V = h(RID \oplus r \oplus t) \quad (19)$$
$$R \rightarrow S : M_1, M_2, M_3, C_i, r, V, T$$

Step 4 : The server (S) checks the validity of *V*, and performs the next stage if valid. If M_2 is valid, the server generates random information *s* and M_4. If not valid, the session is closed. At the verification stage, if the value of M_2, received from the Tag, is equal to $T_r(T_t(x))$, then the server transmits the generated messages M_4, *info*, H_{info} and M_5 to the reader. The server generates a message *info* consisting of *RID* and *data* for his identification credential. these are combined to the XOR operation. Thereafter, the server generates a hash function H_{info} for the *r* and *data*.

$$S : M_2 = ? \; T_{r,t}(x)$$
$$S : S \in \{0, 1\}^l$$
$$S : M_4 = h(ID) \oplus s \oplus r$$
$$S : M_5 = T_{s,t}(x)$$
$$S : info = RID \oplus data \quad (20)$$
$$S : H_{info} = h(data \oplus r)$$
$$S \rightarrow R : M_4, M_5, info, H_{info}$$

Step 5 : The reader computes *data* and checks the validity of H_{info}. If H_{info} is valid, the reader transmits M_4 and M_5 to the tag.

$$R : data = info \oplus RID$$
$$R : H_{info} = ? \; h(data \oplus r) \quad (21)$$
$$R \rightarrow T : M_4, M_5$$

Step 6 : After the tag finds *s* from M_4, it checks the validity of M_5. The tag obtains the *s* via XOR operation using random number *r*, message M_4 and tag's *h*(*ID*). If M_5 is valid, the tag performs the computation below for update of session key *x*

$$T : s = M_4 \oplus H(ID) \oplus r$$
$$T : if \; M_5 = T_{s,t}(x)$$
$$T : x = x \oplus T_{t||s}(x) \quad (22)$$

2.6.2 Attack

The main observations identified in the *Benssalah et al*. method are its logic error in the computation of M_1, and its vulnerability to the tracking attack and to a tag impersonation attack. First, in terms of M_1 generation, information such as *h(ID)*, *r*, *t*, and *ID* all possess the same length, i.e., number of bits. The messages are then generated through XOR operations, but because of the logical error the XOR operation occurs between the 1-m-bit and the 2-m-bit range. Second, the tracking attack uses the weak point of the logical error explained above. The attacker taps the last successful session between the tag and the reader or sends a new request message to the tag. The attacker saves the values of *r* and M_1 generated in step (1) and closes the session.

(1) As mentioned above, if M_1 is generated, the actual bit length is *2m*, expressed in the form shown below.

$$\begin{aligned} M_1 &= h(ID)\oplus((r\oplus t)||(t\oplus ID))\oplus t \\ &= h(ID)\oplus(r\oplus t\oplus t)||(t\oplus ID) \\ &= h(ID)\oplus r||(t\oplus ID) \end{aligned} \quad (23)$$

Next, the attacker can obtain $h(ID)\oplus r$ from the derived message M_1, as well as $h(ID)$ from the already-saved value of r.

(2) The attacker can track the tag using $h(ID)$ or disguise itself.

Lastly, the tag impersonation attack occurs when the attacker impersonates the Tag using the data exposed in the above protocol, and the message as well as the authentication data can be forged using the formulation below.

(1) The attacker taps the last successful session between the tag and the reader or sends a new request message to the tag.

(2) The attacker acquires r, M_1, M_2, M_3, and C_i altogether from the above communication data, closes the session, and ignores the next step.

(3) Using the same method, the attacker obtains r', M_1', M_2', M_3', and C_i

(4) Using the tracking attack explained above, $h(ID)$ is obtained by dividing M_1 by two, and r is freely modulated. Then, the messages also change as described below.

$$\begin{aligned} M_1' &= (h(ID)\oplus r')||(t\oplus ID) \\ M_2' &= T_{\frac{r'}{r}}(M_2) \\ M_2' &= T_{\frac{r'}{r}}(T_{r,t}(x)) \\ M_2' &= T_{\frac{r'}{r}r,t}(x)) \\ M_2' &= T_{r',t}(x) \\ M_3' &= M_3 = x\oplus t \\ c_i' &= c_i \end{aligned} \quad (24)$$

In a similar manner, $M_1'$and M_2', falsified by the attacker, are authenticated by the server as valid, and the Server computes M_4, and M_5 to transmit to the Tag.

3. Security Requirements

A NFC payment service must offer confidentiality, integrity, and user authentication function regarding the data exchanged between the client device and the server [10], and solve the problems of computational efficiency and safety by considering the characteristic of the communication method used for the payment. Moreover, this study also considers the

resistance of the vulnerability of the above-mentioned existing studies (*Cheng et al.*, *Benssalah et al.*). The security requirements of the proposed scheme are as follows.

. Secret Disclosure Attack: The secret value of the tag must not be exposed without physical damage to the tag.

. Mutual Authentication: Mutual authentication must be offered to each individual to show its legitimacy to each other.

. Forward Security: The information exposed at the previous session should not harm the security of the current session.

. Database Storage Efficiency: Additional costs must be avoided by efficiently using the database storage.

. Replay Attack: The attacker cannot use the information from the previous session and disguise itself as the tag or the reader in the current session.

. Tag Identity Confidentiality: The attacker cannot track the location of an individual nor disguise itself as the individual using the personal information of the tag or the reader.

. De-synchronization Attack: It should contain resistance to the de-synchronization attack proposed by Akgun et al. That is, the back-end server will perform key renewal rather than the tag in order to prevent de-synchronization of the session key.

4. Enhanced Mutual Authentication Scheme

In this section, we describe our scheme for light-weight device in low-cost NFC service environments, and show that it satisfies the security requirements specified in section 3. Moreover, our scheme improved the vulnerability of the above-mentioned existing studies. The mutual authentication phase between the server and tag based on our proposed authentication scheme uses enhanced Chebyshev polynomials. Although this scheme is similar to those proposed in previous model, it is comparatively efficient in terms of computational complexity because most of existing vulnerability improved efficiently. Our method was shown to provide very high efficiency in payment environments using the passive communication mode of NFC. The proposed method includes a server authentication phase, key update phase and tag authentication phase. The procedure of each phase is as follows.

4.1 System Parameters

The system parameters in the proposed scheme are as follows.

. Object (T : User or Tag, R: Reader, S: Server)
. g : Positive integer($g < p$)
. p : Prime number($g < p$)

. s, t : The secret information generated by each of the objects ($0 < m < p$)
. x : The current session key to a successful transmission
. x_{old} : The last session key to a successful transmission
. $ID \oplus x$: Search indexing information available in the database
. $T_{*,*}(x)$: generated Chebyshev Polynomial x by secret information *
. ID : ID information of the NFC tag
. $h(ID)$: Hash information of the NFC tag ID
. $\in$: Random Choice operator
. $\oplus$: The bit-wise XOR operation

4.2 User Authentication Phase

Step 1: The Reader (R) generates a random number r and transmits it to the Tag (T).

$R : r \in \{0,1\}^r$

$R \rightarrow T : r$

Step 2: The tag, after receiving r, generates a random number t and compute M_1, M_2, and M_3. M_1 consists of tag ID, random number t and r, and these are combined by XOR operation. M_2 is Chebyshev polynomial based chaotic encryption message encrypted by random number r and t. This message is used as a signature for user authentication in this protocol. At this point, the encrypted information $x \oplus r$ by chaotic cipher provides safety for the M_2 from the tag impersonation attack. Last, M_3 consists of the session key x and random number t between the tag and the server, and these are the XOR operation with each other. If messages have been correctly verified by the server, tag can be trusted by the server. Next, the tag transmits the generated message each message to the reader.

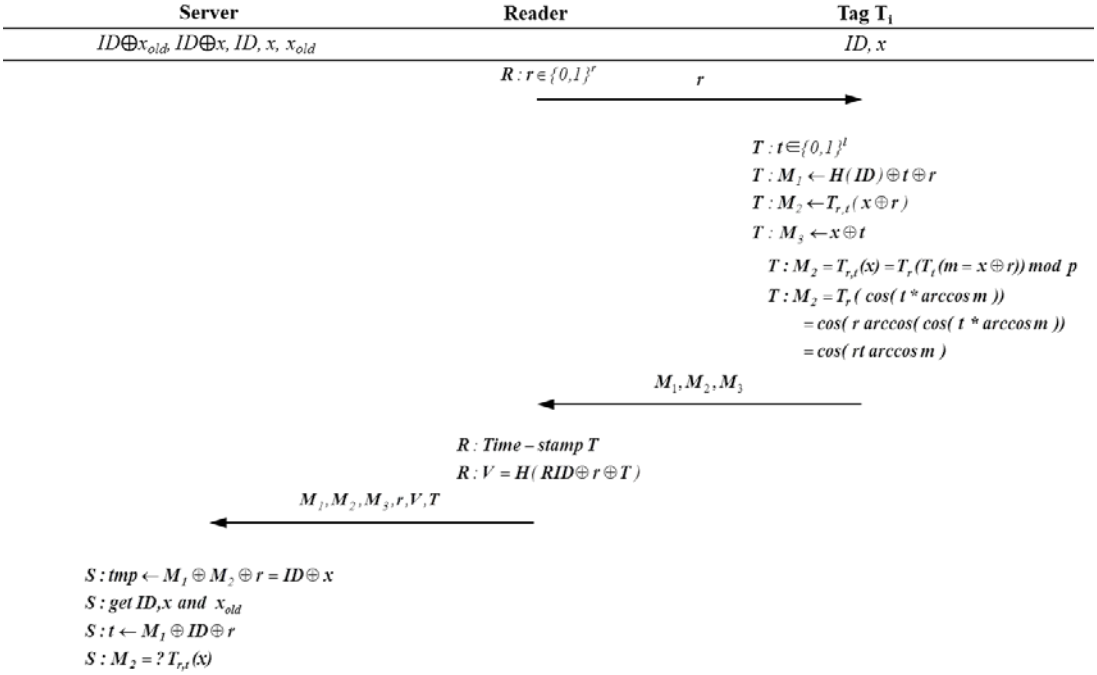


Fig. 2. User authentication phase

$$
\begin{aligned}
&T : t \in \{0,1\}^t \\
&T : M_1 = ID \oplus t \oplus r \\
&T : M_2 = T_{r,t}(x \oplus r) \\
&T : M_3 = x \oplus t \\
&T \rightarrow R : M_1, M_2, M_3
\end{aligned}
\tag{25}
$$

Step 3: The reader, after receiving the messages from the tag, generates the time stamp T and computes the following. Message V generated by reader is a authentication data for verifying the reader by server. V consists of *RID* of reader, random number r and time-stamp T which are performed by a hash function. Later, the server identifies the *ID* of the reader through to validation process using V, r and T. Next, the reader transmits M_1, M_2, M_3, r, V, and T to the server.

$$
\begin{aligned}
&R : \textit{generates time-stamp } T \\
&R : V = h(RID \oplus r \oplus T) \\
&R \rightarrow S : M_1, M_2, M_3, , r, V, T
\end{aligned}
\tag{26}
$$

Step 4: The server (S), after receiving the messages, checks the validity of V. *RID* is unknown for anyone, except for the reader and the server. Therefore, the server generates V_k as much as the number k of the reader using *RID* of all on database. and if valid, computes the value of $M_1 \oplus M_3 \oplus r$ and then, verify the message. The server generates a value $ID \oplus x$ through XOR operation of the M_1, M_3, r. After the server find the same value $ID \oplus x$ through databased index by the number z of tag, and verifies the identity of the tag. The server can obtain current session key x_{new}, last session key x_{old}, current identity data $ID \oplus x$, last identity data $ID \oplus x_{old}$, ID of the tag through identity data $ID \oplus x$ of the tag.

$$
\begin{aligned}
&S : V = ? \; V_{[0,k]} \\
&\textit{For example: } V = ? \; (V_0 = h(RID_0 \oplus r \oplus T)), \\
&\qquad\qquad V = ? \; (V_1 = h(RID_1 \oplus r \oplus T)), \\
&\qquad\qquad \cdots , \\
&\qquad\qquad V = ? \; (V_k = h(RID_k \oplus r \oplus T)) \\
&S : ID \oplus x' = M_1 \oplus M_3 \oplus r \\
&S : ID \oplus x' = ?(ID \oplus x)_{[0,z]} \\
&\textit{For example: } ID \oplus x' = ?(ID_0 \oplus x_0), \\
&\qquad\qquad ID \oplus x' = ?(ID_1 \oplus x_1), \\
&\qquad\qquad \cdots , \\
&\qquad\qquad ID \oplus x' = ?(ID_z \oplus x_z)
\end{aligned}
\tag{27}
$$

Step 5: The server obtains the t via XOR operation using random number r, message M_1 and tag's ID obtained through database indexing, and checks the validity of M_2 through computed $T_r(T_t(x_{new} \oplus r))$ or $T_r(T_t(x_{old} \oplus r))$. As previously described, M_2 is chebyshev polynomial-based chaotic cipher. The server performs a chaotic encryption using the secret key x, random number r and, random number t obtained in step 5 to obtain the signature such as a message M_2. At this point, Server will consider the De-synchronization state between a the tag and the server. first, the server performs a chaotic encryption using current session key x_{new}. At this time, If the failed validation of M_2, the server performs a chaotic encryption using last session key x_{old}.

$S: t = M_1 \oplus ID \oplus r$ (28)
$S: first,\ to\ select\ x_{new},\ then\ M_2 =?\ T_r(T_t(x_{new} \oplus r)) \rightarrow If\ that\ fails,$
$S: second,\ to\ select\ x_{old},\ then\ M_2 =?\ T_r(T_t(x_{old} \oplus r))$

4.3 Key Update Phase

Step 1: Session key x_{new} and x_{old} are only updated after a successful authentication session by server. If the server performs successful verification of the M_2 using the current session key, the server updates to the x_{new} in the x_{old}'s storage, and then generates a new key x_{new}. Conversely, If the server performs successful verification of the M_2 using the last session key, the server determines that the synchronization failed, and generates to current session key x_{new} using last session key x_{old}. At this time, chaotic encryption method is based on the foundation studies proposed by *Lee et al* [1]. We proposed considering processing power of NFC tag. It is capable of utilizing in low-cost NFC tag environments.

$S: If\ x = x_{new},\ then$
$S: x_{old} \leftarrow x_{new}$
$S: x_{new} \leftarrow T_{t||s}(x_{new})$
$S: Else,\ if\ x = x_{old},\ then$
$S: x_{old} \leftarrow x_{old}$
$S: x_{new} \leftarrow T_{t||s}(x_{old})$

4.4 Server Authentication Phase

Step 1: Through the previous steps, If M_2 is valid, the server generates random information s and M_4. Message M_4 consists of tag's ID, random number r and s, and these are combined by XOR operation. Random number s is the information necessary to generate to the x_{new} be used in the tag, and this information can be obtained only the tag and the server know the identity of the tag via message M_4. If not valid then the session is closed.

$S: S \in \{0, 1\}^l$ (29)
$S: M_4 = ID \oplus r \oplus s$

Step 1.1: At the verification step, if M_2, received from the tag equals $T_r(T_t(x_{new} \oplus r))$, then the server generates a Chaotic encryption message M_5 for server authentication. M_5 consists of session key x_{new}, t and s generated by the server. M_5 is chebyshev polynomial based chaotic encryption message encrypted by random number r and s. This message is used as a signature for server authentication in this protocol. At this point, the encrypted information $x \oplus s$ by chaotic cipher provides safety for the M_5 from the server impersonation attack.

$S: if\ M_2 = T_r(T_t(x_{new} \oplus r))$
$S: M_5 = T_{s,t}(x_{new} \oplus s)$ (30)

Step 1.2: At the verification step, if M_2, received from the tag equals $T_r(T_t(x_{old} \oplus r))$, then the server generates a Chaotic encryption message M_5 for server authentication. M_5 consists of session key x_{old}, t and s generated by the server.

$S: \text{if } M_2 = T_r(T_t(x_{old} \oplus r))$
$S: M_5 = T_{s,t}(x_{old} \oplus s)$ (31)

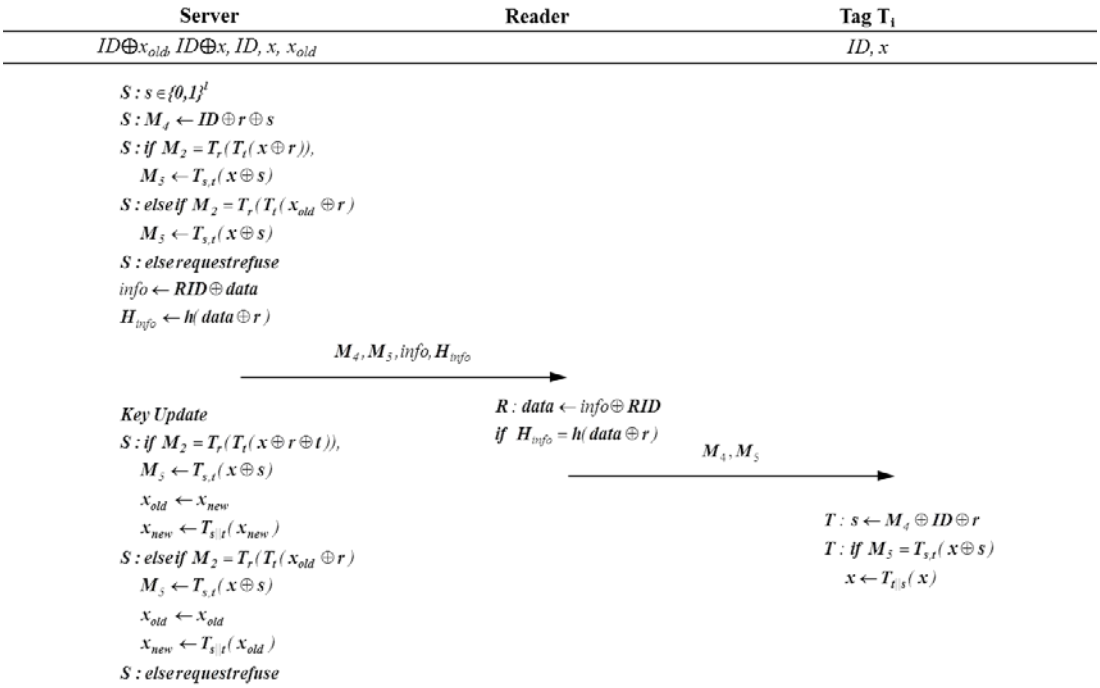


Fig. 3. key update and server authentication phase

Step 2: The server generates a message *info* consisting of *RID* and *data* for his identification c redential. these are combined to the XOR operation. Thereafter, the server generates a hash function H_{info} for the *r* and *data* to provide the integrity of the message *info*. The server transmits the generated messages M_4, M_5, *info*, and H_{info} to the reader.

$$S : info = RID \oplus data$$
$$S : H_{info} = h(data \oplus r) \quad (32)$$
$$S \rightarrow R : M_4,\ M_5,\ info,\ H_{info})$$

Step 3: The reader computes *data* using XOR operation between *info* and *RID*. After, the read er checks the validity of H_{info} using hash function and XOR operation between *data* and rando m number *r*. If H_{info} is valid, the reader transmits M_4 and M_5 to the tag.

$$R : data = info \oplus RID$$
$$R : If\ \ H_{info} = h(data \oplus r),\ then \quad (33)$$
$$R \rightarrow T : M_4,\ M_5,$$

Step 4: The tag, after receiving the messages from the server, computes *s* and $T_s(T_t(x \oplus r))$, and checks the validity of M_5. The tag obtains the s via XOR operation using random number r, message M_4 and tag's ID. Then, the tag performs the computation below for update of session key *x*. In this step, we do not distinguish the session key *x*. In view of the tag, Classification of x_{new} or x_{old} is meaningless. Because, session key *x* is fixed by the synchronization process.

$$T : s = M_4 \oplus H(ID) \oplus r$$
$$T : If\ M_5 = T_{s,t}(x \oplus r),\ then \quad (34)$$
$$T : x \leftarrow T_{t||s}(x)$$

5. Analysis of Proposed Scheme

5.1 Secret Disclosure Attack

Our proposed scheme prevents the exposure of the tag's secret value unless there is physical damage. When the data including the *x* information is exchanged from communications, it is secured via a proprietary algorithm using XOR and a Chebyshev chaotic map to prevent the exposure of the value of *x* using the information on the public channel. In addition, after each successful authentication session, *x* is re-initialized by the key update step, providing a high level of security. Moreover, the attackers cannot execute a secret exposure impersonation attack using the exposed information on public channels. This situation is characterized when the weak points of the existing methods are all enhanced and the attacker cannot access *t*, *s*, or *ID* information to find the x value hidden by the XOR operations. As follows, our proposed method can be improved to the Chebyshev vulnerability by inverse function. In the following, we were comparing our enhanced mutual authentication method with vulnerability of previous methods. The case of traditional methods, it is capable of Tracking Attack for M_1. As mentioned in formula (23), if M_1 is generated, the actual bit length is 2-m, expressed in the form shown below.

$$M_1 = h(ID) \oplus ((r \oplus t) || (t \oplus ID)) \oplus t$$
$$= h(ID) \oplus (r \oplus t \oplus t) || (t \oplus ID)$$
$$= h(ID) \oplus r || (t \oplus ID)$$

After, An attacker can obtain the $H(ID) \oplus r$ from the message. Therefore, the attacker is possible to obtain M_1' from the M_1.

$$M_1 = (\, h(ID) \oplus \mathbf{r} || (t \oplus ID) \rightarrow M_1' = h(ID) \oplus \mathbf{r'} || (t \oplus ID)$$

However, in the case of the proposed method, the attacker is not possible to obtain *H(ID)*, *r*, *t* through the acquired M_1. Next, In the case of impersonation attack of attacker, he obtains M_2, generate forged signature. In the case of traditional methods, the attacker can forge the message M_1. Also, he is able to forge a signed message M_2 through the public information *r* and inverse function. However, the proposed scheme is not possible attacks using the inverse because it contains *x*, *r*, *t* in the signature information.

Table 1. Compared of traditional methods and proposed method

Traditional scheme's M_2	*Our scheme's* M_2
$M_2' = T_{\frac{r'}{r}}(M_2)$	$M_2' = T_{\frac{r'}{r}}(M_2)$
$M_2' = T_{\frac{r'}{r}}(T_{r,t}(x))$	$M_2' = T_{\frac{r'}{r}}(T_{r,t}(x \oplus r \oplus t))$
$M_2' = T_{\frac{r'}{r}r,t}(x))$	$M_2' = T_{\frac{r'}{r}r,t}(x \oplus r \oplus t))$
$M_2' = T_{r',t}(x) \leftarrow Verifiable$	$M_2' = T_{r',t}(x \oplus r \oplus t) \leftarrow error$

5.2 Mutual Authentication

Our proposed scheme can also offer mutual authentication. First, we prove that the tag provides authentication, showing that it is resisting the impersonation attack. Also, we prove that the reader is authenticated, showing that the algorithm is resistant to the reader-masquerade attack. Because each entity needs to confirm the *ID* information of the other side before allowing any transaction, it is necessary to prevent "spoofing" or masquerading of the server or the tag. In other words, it is necessary to block the attacker from manipulating the sensitive data of a legal tag, and from executing a masquerade attack. Therefore, the mutual authentication protocol is important in any case to satisfy both requirements. In the proposed scheme, the server determines the tag's validity by verifying the values of M_1, M_2, M_3 received from the tag. In particular, the server computes the values shown below on the basis of the Chebyshev chaotic map hard problem and the XOR operations, the validity of the message is determined via a database record search, and the tag is subsequently authenticated. According to the Chebyshev chaotic map definition 1, 2, 4, it is calculated as shown below.

$$
\begin{aligned}
&ID \oplus x = M_1 \oplus M_3 \oplus r \\
&ID \oplus x' =? ID \oplus x \leftarrow \textit{checks the validity of message via database record search} \\
&t \leftarrow M_1 \oplus H(ID) \oplus r \\
&M_2 =? T_r(T_t(x \oplus r)) \\
&\quad = T_r(T_t(x \oplus r)) = \cos(t \arccos(\cos(r \arccos x \oplus r))) \\
&\quad = \cos(tr \arccos x) \\
&\quad = \cos(rt \arccos x) \\
&\quad = \cos(r \arccos(\cos(t \arccos x \oplus r))) \\
&\quad = T_r(T_t(x \oplus r))
\end{aligned}
\tag{35}
$$

In a similar manner, the tag verifies the validity of values of M_4 and M_5 after receiving them from the server. The tag obtains the nonce value s from M_4, which is then used to compute M_5, generated by the Chebyshev chaotic map. If the received value of M_5 agrees with the computation, then the tag can trust the server. In this way, we offer mutual authentication.

$$
\begin{aligned}
&s \leftarrow M_4 \oplus H(ID) \oplus r \\
&M_5 =? T_{s,t}(x \oplus r) \\
&\quad = T_s(T_t(x \oplus r)) = \cos(t \arccos(\cos(s \arccos x \oplus r))) \\
&\quad = \cos(ts \arccos x) \\
&\quad = \cos(st \arccos x) \\
&\quad = \cos(s \arccos(\cos(t \arccos x \oplus r))) \\
&\quad = T_s(T_t(x \oplus r))
\end{aligned}
\tag{36}
$$

5.3 Forward Security

If the secret information of the messages used in the previous session is exposed by the attacker, it should not threaten the current session. Assume that the attacker knows the tag's secret information, *x*, *ID*, etc. Also assume that the attacker cannot know the other random numbers used by the server and the tag to configure the message, and the secret key

information x is re-initialized for every session, and then the attacker cannot pose any risk to the current session.

5.4 Data Storage Efficiency

Our proposed scheme improves the storage efficiency of the server's database. In the case of the existing methods, other than ID, x_{new}, x_{old}, c_{new}, c_{old}, $T_{new}(*)$, and $T_{old}(*)$, values must be generated to identify the tag and declare an additional variable table. Our proposed scheme deletes unnecessary parameters based on the *Cheng et al.* scheme and thus increases the storage efficiency of the database. This can offer extremely high efficiency and increased scalability in an environment of a large-scale application business model.

5.5 Replay Attack

The attacker would use the message from the previous protocol session in order to impersonate a legal tag or a legal reader. That is, M_1, M_2, M_3, M_4, M_5, and r on the public channels are reused. However, in our proposed scheme, the major nonce values of t and s are secured by the Chebyshev chaotic map hard problem, and the messages from the previous session cannot be reused in another session.

5.6 Tag Identity Confidentiality (Tracking Attack)

We must protect the tag ID's and other parameters from the attacker to provide confidentiality. The attacker can obtain M_1, M_2, M_3, M_4, and M_5 on the pubic channels, but cannot reach the ID using these values. However, if the attacker does not know the tag's ID and the shared secret key, it cannot configure a valid authentication answer. Therefore, our scheme can provide safety from the tracking attack and impersonation attack.

5.7 De-synchronization Attack

The scheme proposed herein performs key update to provide resistibility to de-synchronization attack. x_{new} and x_{old} are only updated after a successful authentication session. That is, the proposed scheme saves the secret key of the previous session to prevent the de-synchronization attack. This enables protection of each individual from the de-synchronization attack.

5.8 Performance evaluation

A general RFID tag, including NFC, is a highly limited device in terms of its computational and storage requirements. Therefore, when designing an RFID system protocol, a low cost encryption implementation must be considered. In our proposed scheme, the tag performs PRNG, the creation of the Chebyshev chaotic map, XOR operations, and concatenation operations. Recently, the Chebyshev polynomial has been proposed as the

authentication system for a smart card, The Diffie–Hellman key agreement protocol as well as the RFID system [7,10,13,14]. [14] show the feasibility of the key agreement protocol based on the Chebyshev polynomial for smart cards, and [13] also proved the feasibility of a low-cost RFID system also based on the Chebyshev polynomial. Other than these, there have been studies to improve configuration issues, efficiency, time, and cost [14-17]. In the early authentication protocol studies [4, 5], simple XOR operations and PRNG operations were used to provide authentication between individuals, but in recent studies [6,8-11], safety has been further improved by providing authentication of the messages using the lightweight Chebyshev polynomial. In particular, the proposed technique not only enhances the efficiency of the database storage but also reduces the computational requirements of the tag. In addition, its vulnerability to the tracking attack and the tag impersonation attack, as shown in the existing techniques, was decreased as well. Table 1 compares the security functions offered by the previously proposed protocols to our protocols. In addition, Table 3 compares performance of the two protocols. Our proposed protocol provides all security functions required by the security requirements.

Table 2. Analysis of Proposed Scheme

		[4]	[5]	[6]	[10]	Proposed Scheme
Secret disclosure attacks		✓	×	×	×	✓
Mutual authentication		✓	✓	✓	✓	✓
Forward security		✓	×	×	×	✓
Replay attack		✓	✓	×	×	✓
Tag identity confidentiality		×	×	×	×	✓
De-synchronization attack		✓	✓	×	✓	✓
Computation	**Tag**	3S+4H+1R+8X	1R+6P	$2T_n$+1R+6X	$4T_n$+1R+9X	$3T_n$+1R+7X
	Reader	1R+2X	1R+1H+3X	1R	1R+2H+4X	1R+2H+4X
	Server	3SRS+13X	1R+4P+20X	$2T_n$+1R+7X	$4T_n$+1R+1H+5X	$2T_n$+1R+1H+10X
Key search complexity		*O*(1)	*O*(1)	*O*(1)	*O*(1)	*O*(1)
Database Storage efficiency (Server side)		9m	9m	6m	8m	6m
Communication rounds		5	5	5	5	5
Crypto primitive		X,P	X,P	T,X,C	T,X,C	T,X,C

H: hash function; M: message length; C: concatenation; P: pseudo random number generation; R: random number generation; SRS: Square root solving; S: modular squaring; T_n: Chebyshev polynomials; X: XOR

6. Conclusions

This study proposed an authentication technique that provides the mutual authentication between the NFC tag and the reader (or server) using the chaotic map. The chaotic map cannot easily produce the output and it appears random. Such a characteristic is similar to what is required for a cryptographic system, as the most chaotic crypto systems require a high level of computational power and, therefore, it was impossible to use with a lightweight system. Recently, however, various studies have shown that the chaotic crypto system performs with computational time similar to HASH, i.e., *O(n)*. Thus, these studies on the usage of a chaotic map have received considerable attention. In this paper, we proposed a new mutual authentication system using chaotic map. This method has the same traffic as the conventional method. However, providing a safety against the vulnerability of the conventional method. In particular, unlike hash functions and XOR operations, the Chebyshev polynomial based chaotic cryptosystem is possible to provide a high cryptographic strength. Also, our proposed scheme provide to high quality security from a variety of security threats. Accordingly, the proposed method can provide a high efficiency in light weight environment. Finally, it is determined to be an effective service provided by the passive communication mode of the NFC.

References

[1] Cheng-Chi Lee, "A simple key agreement scheme based on chaotic maps for VSAT satellite communications," *International Journal of Satellite Communications and Networking*, Vol. 31, Issue 4, pp. 177–186, 2013. Article (CrossRef Link).

[2] G. J Fee, M. B. Monagan, "Cryptography using Chebyshev polynomials," *Citeseer*, 2004. Article (CrossRef Link).

[3] D Xiao, X Liao, S Deng, "A novel key agreement protocol based on chaotic maps," *Information Sciences*, Vol. 177, Issue 4, pp. 1136–1142, 2007. Article (CrossRef Link).

[4] 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*, Vol. 37, Issue 12, pp 7678–7683, December 2010. Article (CrossRef Link).

[5] Yoon E, "Improvement of the securing RFID systems conforming to EPC class 1 generation 2 standard," *Expert Systems with Applications*, Vol. 39, Issue 1, pp. 1589-1594, January 2012.. Article (CrossRef Link).

[6] C.C. Chang, C.Y. Sun, "A Secure and Efficient Authentication Scheme for E-coupon Systems," *Wireless Pers Coomun*, Vol. 77, Issue 4, pp. 2981–2996, 2014. Article (CrossRef Link).

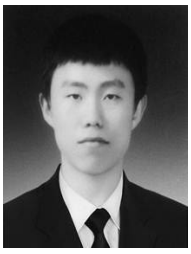
[7] Zi-Yao Cheng, Yun Liu, Chin-Chen Chang, Shih-Chang Chang, "Authenticated RFID security mechanism based on chaotic maps," *Security and Communication Networks*, Vol. 6, Issue 2, pp. 247–256, February 2013. Article (CrossRef Link).

[8] M. Akgün, M. U. Caglayan, "Weaknesses in a Recently Proposed RFID Authentication Protocol," *IACR Cryptology ePrint Archive*, 2013. Article (CrossRef Link).

[9] M. Akgün, M. U. Caglayan, "Vulnerabilities of RFID Security Protocol Based on Chaotic Maps," in *Proc. of Network Protocols (ICNP), 2014 IEEE 22nd International Conference on*, 21-24 Oct. 2014. Article (CrossRef Link).

[10] Benssalah M, Djeddou M, Drouiche K., "Security enhancement of the authenticated RFID security mechanism based on chaotic maps," *Security and Communication Networks*, Vol. 7, Issue 12, pp. 2356–2372, December 2014. Article (CrossRef Link).

[11] M. Akgun, A. O. Bayrak, M. U. Caglayan, "Attacks and improvements to chaotic map-based RFID authentication protocol," *Security and Communication Networks*, Vol. 8, Issue 18, pp. 4028–4040, December 2015. Article (CrossRef Link).
[12] Di Xiao, Xiaofeng Liao, K.W. Wong, "An efficient entire chaos-based scheme for deniable authentication," *Chaos, Solitons & Fractals*, Vol. 23, Issue 4, pp. 1327–1331, February 2005. Article (CrossRef Link).
[13] Yujun Niu XW, "An anonymous key agreement protocol based on chaotic maps," *Communications in Nonlinear Science and Numerical Simulation*, Vol. 16, Issue 4, pp. 1986–1992, April 2010. Article (CrossRef Link).
[14] Guo C, Chang CC, "Chaotic maps-based password authenticated key agreement using smart cards," *Communications in Nonlinear Science and Numerical Simulation*, Vol. 18, Issue 6, pp. 1433–1440, June 2013. Article (CrossRef Link).
[15] Wang XY, Zhao JF, "An improved key agreement protocol based on chaos," *Communications in Nonlinear Science and Numerical Simulation*, Vol. 15, Issue 12, pp. 1052-4057, December 2010. Article (CrossRef Link).
[16] Bergamo P, D'Arco P, De Santis A, Kocarev L, "Using time-stamp to improve the security of a chaotic maps based key agreement protocol," *Information Sciences*, Vol. 178, Issue 6, pp. 1598–1602, 2008. Article (CrossRef Link).
[17] Fateman RJ, "Lookup tables, recurrences, and complexity," in *Proc. of ISSAC '89 Proceedings of the ACM-SIGSAM 1989 international symposium on Symbolic and algebraic computation*, pp. 68-73, Portland, Oregon, USA, July 17-19, 1989. Article (CrossRef Link).



Sung-Wook Park received the B.S. and M.S. degrees in Depart of Computer Software Engineering from Soonchunhyang University, Korea, in 2011 and 2013, respectively. He is now a Ph.D. candidate in Department of Computer Software Engineering from Soonchunhyang University, Korea. His research interests include NFC Security, NTRU Cryptography, Ultra Lightweight Cryptography, etc.

Im-Yeong Lee is corresponding author. He received the B.S. degrees in Department of Electronic Engineering from Hongik University, Korea, in 1981 and the M.S. and Ph.D. degrees in Department of Communication Engineering from Osaka University, Japan, in 1986 and 1989, respectively. From 1989 to 1994, he had been a senior researcher at ETRI (Electronics and Telecommunications Research Institute), Korea. Now he is a professor in Department of Computer Software Engineering from Soonchunhyang University, Korea. His research interests include Cryptography, Information theory, Computer & Network security.