天主教輔仁大學圖書資訊學系碩士班碩士論文

指導老師：李正吉 博士

應用擴充式混沌映射於使用者認證和金鑰協議機制之研究

The Study of User Authentication and Key Agreement Protocols Based on Extended Chaotic Maps

研究生：許哲維 撰

中華民國102年7月

誌謝 (ACKNOWLEDGEMENTS)

首先誠摯感謝指導教授李正吉博士的悉心教導與照顧，使我在短短的兩年內得以窺見資訊安全領域的深奧，並且不時的討論並指點我正確的方向，使我在這些兩年中獲益匪淺。尤其老師對學問的嚴謹和事務上的態度更是我輩應該學習的典範。

另外，本論文的完成亦得感謝謝建成教授與李俊達教授的大力協助。因為有你們的審閱及幫忙，使得本論文能夠更加完善而嚴謹。

兩年的日子中，在研究室裡共同的生活點滴，不管是學業上的討論或是閒暇時間的談天說地，衷心感謝各位學長姐們、同學們與學弟妹們的共同砥礪，你/妳們的陪伴讓兩年的研究生生活變得燦爛無比、充滿歡笑。

感謝秉憲學長在研究上對我的幫助，也感謝彥銘同學這兩年間的幫忙與照顧，恭喜我們終於順利走過這兩年。另外還有彥宏學弟，詩婷、姿穎與佩娟學妹，感謝妳們平時的照顧與口試時的幫忙。當然，也要感謝那些在背後默默支持我的人，你/妳們的支持是我前進的動力。

最後要感謝我的家人，從上大學時離家北上念書到現在，六年的時光終於順利從大學與研究所畢業，沒有你們的支持與鼓勵就沒有現在的我，非常感謝你們。

感謝這六年間所有幫助過我的人，衷心地感謝你/妳們。

中文摘要

由於計算機網路的快速發展與成長，現今有許多人透過公開網路使用電子郵件或是即時通軟體來進行溝通或是交換訊息。儘管如此，計算機網路仍是一個不安全的分享平台，由於其可能遭受到某些安全性攻擊，像是惡意攻擊者的監聽、密碼猜測攻擊或是偽造攻擊來欺騙其他合法使用者。因此，如何設計一個安全的通訊協議機制使得使用者可以安全地溝通、交換訊息成為一個重要的議題。在過去數十年來，許多研究被發表來提供一個安全地計算機網路環境，像是使用智慧卡的遠端使用者認證方案、基於密碼的三方認證金鑰交換協定與應用於多重伺服器環境的金鑰協議機制。這些機制提供使用者認證或是金鑰協議來確保資訊的安全性，使得使用者可以認證彼此的合法性，並且藉由協議出一把金鑰來加密與解密所要傳輸的訊息，在公開網路上安全地溝通、交換訊息。

在本研究中，我們將分析與探討三種使用者認證與金鑰協議的機制，並且分別指出他們機制的安全性漏洞。除此之外，我們也提出基於擴充式混沌映射的改善機制來解決上述的安全性漏洞。根據安全性與效率分析說明，我們所提出的機制相較於他們提出來的機制顯得更加安全及更有效率。

關鍵字：匿名性，生物特徵，混沌映射，金鑰協議，相互認證，智慧卡，三方認證。

ABSTRACT

Due to the rapid development and growth of computer network, many people use email or messenger software to communicate with each other through the public network. Nevertheless, the computer network is an insecure shared platform since it may vulnerable to some security attacks such as eavesdropping, password guessing attack, and impersonation attack. Therefore, how to design a secure communication protocol that users can securely communicate with each other becomes an important issue. Over the past few decades, many researchers have been proposed to provide a secure computer network environment, such as remote user authentication protocol using smart cards, three-party password-based authenticated key exchange protocol, and key agreement protocol in multi-server environments. These protocols provide user authentication or key agreement to guarantee the information security that users can authenticate with each other and securely exchange messages over a public network by using the shared session key to encrypt and decrypt the secure information.

In this study, we will analyze three user authentication and key agreement protocols and point out the security flaws of their protocols. Besides, we also propose our protocols based on extended chaotic maps to remedy these security weaknesses of their protocols. As compared with their protocols, the security and performance analysis show that our proposed protocols are more secure and efficient than theirs.

Keywords: Anonymity, Biometric, Chaotic maps, Key agreement, Mutual authentication, Smart card, Three-party authentication.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1 Introduction

1.1 Research Motivation

Nowadays, the password-based authentication protocol is an essential technique which used to identify the validity of a remote user in client/server system [12, 31, 35, 64, 69]. But it has a major problem in that humans are not experts in memorizing text strings. After that, cryptographic secret keys and passwords which are used in remote user authentication protocols have been proposed to solve the mentioned problems. Nevertheless, it still have some problems, such as the cryptographic secret keys and passwords cannot provide non-repudiation. In order to solve the mentioned problems, the biometric-based remote user authentication protocols have been proposed by researches [26, 44, 51]. In 2011, Das proposed an efficient biometric-based remote user authentication protocol using smart cards [9] to remedy the security flaws in Li and Hwang’s protocol [44]. Unfortunately, we found that the Das’s protocol was vulnerable to privileged insider attacks, off-line password guessing attacks and also cannot provide user anonymity. Hence, we will propose an improved protocol to solve the weaknesses in his protocol.

Due to the rapid development and growth of computer networks, many remote password authentication protocols have been created and well received because of their simple implementation, easy operation, and low cost [46, 47, 59, 70]. Recently, the focus has been on protocols for multi-server environments that run on smart cards. These protocols typically count on the nonce or timestamp to provide protection against the replay attack. But these protocols have some security issues such as disturbance in clock synchronization and vulnerability to the man-in-the-middle attack. In order to solve the mentioned problems, Tsaur et al. proposed a multi-server authentication protocol with key agreement [70] and it used the self-verified timestamp where the timestamp is verified by the timestamp creator. However, we found out that Tsaur et al.'s protocol still has the following security flaws: (1) it cannot resist the privileged insider attack; (2) it cannot resist the known-plaintext attack; (3) it is unable to provide user anonymity; (4) it does not provide perfect forward secrecy. Therefore, we will propose an improved protocol to remedy the security flaws in their protocol.

In order to guarantee the security of secret keys which are exchanged over the insecure public network, some related protocols [6, 7, 40, 41, 49, 61] have been proposed by researchers, such as Password-Authenticated Key Exchange (PAKE) protocol. In 1992, Bellovin and Merritt proposed the first PAKE protocol [2] which allows two parties to keep one identical memorable password to agree on a common session key over the insecure public network [16, 53, 54, 69]. After a decade, many related protocols such as the three-party PAKE [32, 33, 40, 41, 61, 81] also have been proposed. However, some of the three-party PAKE protocols are not secure or efficient enough to be used in practice. Recently, Wu et al. [78] proposed a three-party password-based authenticated key exchange protocol to remedy the security flaws in Huang's protocol [19]. Nevertheless, Wu et al.'s protocol had many exponential computations, which required the highest computational complexity and it also could not provide user anonymity. Hence, we will propose an improved protocol to enhance the security and efficiency of the Wu et al.'s protocol.

Over the past decades, much research has been proposed to design secure communication protocols based on chaotic systems [8, 13, 27, 77]. In order to design a secure, practical, and can be used for both encryption and digital signature's public-key algorithm, Kocarev and Tasev [28] proposed a public-key encryption algorithm based on Chebyshev chaotic maps in 2003. Unfortunately, Bergamo et al. [3] pointed out that Kocarev and Tasev’s protocol [28] is insecure since an adversary can efficiently recover the plaintext from a given ciphertext. In order to remedy this weaknesses, Zhang proposed a protocol [86] and proved that the semigroup property holds for Chebyshev polynomials defined on interval $(-\infty, +\infty)$, which can enhance the property. After that, much work has been done by introducing chaotic maps into the design of symmetric encryption protocols [65, 73, 75], S-boxes [74], biometric-based remote user authentication [37], and hash functions [79, 80]. In this study, we will proposed three improvements based on extended chaotic maps to remedy the security flaws and enhance the efficiency of the Das’s, Tsaur et al.’s, and Wu et al.’s protocols.

1.2 Research Subjects

In this study, we focus on the above three user authentication and key agreement protocols. The first protocol is biometric-based remote user authentication protocol using smart cards. Das proposed an efficient biometric-based remote user authentication protocol using smart cards [9] to remedy the security flaws in Li and Hwang’s protocol [44]. Unfortunately, we found that Das’s protocol was vulnerable to privileged insider attacks, off-line password guessing attacks and also cannot provide user anonymity. Therefore, how to resist the privileged insider attacks and off-line password guessing attacks is the one of our research subjects.

The second one is Password-Authenticated Key Exchange (PAKE) protocol. Wu et al. [78] proposed a three-party password-based authenticated key exchange protocol to solve the security problems in Huang’s protocol [19]. However, Wu et al.’s protocol had many exponential computations, which required the highest computational complexity and it also could not provide user anonymity. Thus, to provide user anonymity and enhance the efficiency is the one of subjects we have to research.

The third one is user authentication and key agreement protocol for multi-server environments. Recently, the focus has been on protocols for multi-server environments that run on smart cards. These protocols typically count on the nonce or timestamp to provide protection against the replay attack. But these protocols have some security issues such as disturbance in clock synchronization and vulnerability to the man-in-the-middle attack. In order to solve these problems, Tsaur et al. proposed a multi-server authentication protocol with key agreement in 2012 [70], and they claimed that their protocol could effectively achieve password-authenticated key agreement while getting around the technical difficulty of implementing clock synchronization in multi-server environments. Nevertheless, we found their protocol still has the following security flaws: (1) it cannot resist the privileged insider attack; (2) it cannot resist the known-plaintext attack; (3) it is unable to provide user anonymity; (4) it does not provide perfect forward secrecy. Hence, a more secure protocol is the one of our research subjects.

1.3 Thesis Organization

The remainder of this thesis is organized as follows. In Chapter 2, we will introduce Das's biometric-based remote user authentication protocol using smart cards and our improved protocol. Then, we will introduce our three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps in Chapter 3. In Chapter 4, we shall review Tsaur et al.'s multi-server authentication protocol with key agreement and present our improved protocol based on extended chaotic maps. Finally, our conclusion will be in Chapter 5.

Chapter 2 A Secure Biometric-based Remote User Authentication with Key Agreement Protocol Using Extended Chaotic Maps

Recently, biometric-based remote user authentication protocols along with passwords have drawn considerable attention in research. In 2011, Das proposed an improvement on an efficient biometric-based remote user authentication protocol using smart cards and claimed his protocol could resist various attacks. However, there are some weaknesses in Das's protocol such as the privileged insider attack and the off-line password guessing attack. Besides, Das's protocol also cannot provide user anonymity. To overcome these weaknesses, we shall propose a secure biometric-based remote user authentication with key agreement protocol using extended chaotic maps. The proposed protocol not only can resist the above-mentioned attacks, but also provide user anonymity.

2.1 Preliminaries

With regard to the client/server system, the password-based authentication protocol is an essential technique used in order to identify the validity of a remote user [12, 31, 35, 64, 69]. Sun et al. [66] pointed out password-based authentication protocols have a major problem in that humans are not experts in memorizing text strings. Therefore, most users would probably choose easy-to-remember passwords even if they know the passwords might be unsafe. In 2005, Hwang-Liu [21] and Lee-Chiu [36] proposed their traditional remote identity-based authentication protocols respectively. The security of their protocols is only based on the passwords. Consequently, the adversary can use brute force attacks or dictionary attacks to break the passwords if users select weak passwords [33, 38, 43, 63]. In order to solve this problem, cryptographic secret keys and passwords are used in remote user authentication protocols. Unfortunately, the cryptographic secret keys and passwords still have some problems such as the use of long and random keys which are difficult to memorize so that the keys must be stored somewhere, and maintaining the long cryptographic keys is expensive. It also cannot provide non-repudiation since the keys may be forgotten, lost or they may be shared with other people, there is no way to know who the actual user is.

Recently, some biometric-based remote user authentication protocols have been proposed by researches [26, 44, 51]. The biometric system is basically a pattern recognition system which operates by obtaining biometric data from an individual, extracting a feature set from the obtained data and comparing this feature set with the template set in the database [23, 45, 57, 62]. Das [9] pointed out that the biometric keys have some advantages as follows:

- Biometric keys cannot be lost or forgotten.
- Biometric keys are very difficult to copy or share.
- Biometric keys are extremely hard to forge or distribute.
- Biometric keys cannot be guessed easily.
- Biometric keys are not easy to break.

As mentioned above, biometric-based remote user authentication protocols are more reliable and secure than traditional password-based remote user authentication protocols. In 2010, Li and Hwang [44] proposed an efficient biometric-based remote authentication protocol using smart cards. After that, Das [9] pointed out that Li and Hwang’s protocol has some flaws and proposed an improvement of Li and Hwang’s protocol to remedy their flaws. Nevertheless, we found that Das’s protocol was vulnerable to privileged insider attacks, off-line password guessing attacks and also cannot provide user anonymity. To remedy these weaknesses, we propose a secure biometric-based remote user authentication with key agreement protocol using extended chaotic maps. The proposed protocol based on chaos theory can allow the user to anonymously communicate with the server and provide mutual authentication between user and server. The security and performance analysis show that the proposed protocol has low computation and communication cost, and also can resist these attacks which was found in Das’s protocol.

2.2 Related Work

In this section, we shall briefly describe the concept of Chebyshev polynomial which used in Chapter 2, Chapter3, and Chapter 4.

2.2.1 Chebyshev Chaotic Maps

Some basic concepts about the Chebyshev polynomial [58] are as follows. The Chebyshev polynomial $T_n(x)$ is a polynomial in x of degree n. Let n be an integer, and let x be a variable taking value over the interval $[-1, 1]$. The Chebyshev polynomial $T_n(x)\colon [-1,1] \to [-1,1]$ is defined as follows:

$$T_n(x) = \cos(n \cdot \arccos(x))$$

The recurrence relation of the Chebyshev polynomial is defined as

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

where $T_0(x) = 1$ and $T_1(x) = x$.

As we know, $\cos(x)$ and $\arccos(x)$ are trigonometric functions [3]. They are defined as $\cos: R \to [-1, 1]$ and $\arccos: [-1, 1] \to [0, \pi]$, respectively. Here are some example Chebyshev polynomials as follows:

$$T_2(x) = 2x^2 - 1$$

$$T_3(x) = 4x^3 - 3x$$

$$T_4(x) = 8x^4 - 8x^2 + 1$$

$$T_5(x) = 16x^5 - 20x^3 + 5x$$

Chebyshev polynomials exhibit two important features [14, 35]: the semigroup property and the chaotic property.

(1) The semigroup property:

$$\begin{aligned} T_r\big(T_s(x)\big) &= \cos(r\cos^{-1}(\cos(s\cos^{-1}(x)))) \\ &= \cos(rs\cos^{-1}(x)) \\ &= T_{sr}(x) \\ &= T_s\big(T_r(x)\big) \end{aligned}$$

Here, r and s are positive integer numbers and $x \in [-1, 1]$.

(2) The chaotic property:

When the degree $n$satisfies the requirement of $n > 1$, the Chebyshev polynomial map $T_n(x): [-1, 1] \to [-1, 1]$ of degree n is a chaotic map with its invariant density being $f^*(x) = 1/(\pi\sqrt{1 - x^2})$ for positive Lyapunov exponent $\lambda = \ln n > 0$.

Zhang further broadened the range of the semigroup property by proving that the semigroup property holds for Chebyshev polynomials defined on interval $(-\infty, +\infty)$ [86] as follows:

$$T_n(x) \equiv \left(2xT_{n-1}(x) - T_{n-2}(x)\right) \bmod p$$

where $n \geq 2$, $x \in (-\infty, +\infty)$, and p is a large prime number that $p = 2^{130} + 7$. Evidently,

$$T_r\left(T_s(x)\right) \equiv T_{sr}(x) \equiv T_s\left(T_r(x)\right) \bmod p$$

so the semigroup property still holds and the enhanced Chebyshev polynomials also commute under composition.

The Chebyshev polynomial poses the following two problems [16, 37], which are assumed to be difficult to handle within polynomial time:

(1) Given two elements x and y, the task of the discrete logarithm problem (DLP) is to find the integer r, such that $T_r(x) = y$.

(2) Given three elements x, $T_r(x)$, and $T_s(x)$, the task of the Diffie-Hellman problem (DHP) is to compute the element $T_{rs}(x)$.

2.3 Review of Das's Protocol

In this section, we describe Das's protocol [9]. The notations throughout Das's protocol are summarized in Table 2.3.1.

Table 2.3.1 The notations used in Das's protocol

Notation	*Definition*
C_i	client
R_i	trusted registration center
S_i	server
PW_i	password shared between C_i and S_i
ID_i	identity of the user C_i
B_i	biometric template of the user C_i
$h(\cdot)$	a secure one-way hash function
X_s	a secret information maintained by the server
R_c	a random number chosen by C_i
R_s	a random number chosen by S_i
$A \parallel B$	data A concatenates with data B
$A \oplus B$	XOR operation of A and B

There are four phases in Das's protocol including the registration phase, login phase, authentication phase, and password change phase. Das's protocol uses the biometric template pattern matching to perform the user's biometric verification [23]. The user's biometric will be matched against the template pattern stored in the system when the user inputs his/her biometric template. The user will pass the biometric verification if there is a match. We explain the details of each phase as follows.

2.3.1 Registration Phase

When the remote user C_i wants to access the system, as shown in Figure 2.3.1, he/she needs to perform the following steps:

(1) The user inputs his/her personal biometric B_i on a specific device and offers his/her password PW_i and the identity ID_i of the user to the registration center R_i in person.

(2) The registration center R_i computes the following

$f_i = h(B_i)$,

$r_i = h(PW_i) \oplus f_i$,

$e_i = h(ID_i \parallel X_s) \oplus r_i$.

X_s is a secret information generated by the server and is not disclosed to any others for all secure future communications.

(3) R_i embedded $(ID_i, h(.), f_i, e_i, r_i)$ in the user's smart card and sends the card to the user C_i via a secure channel.

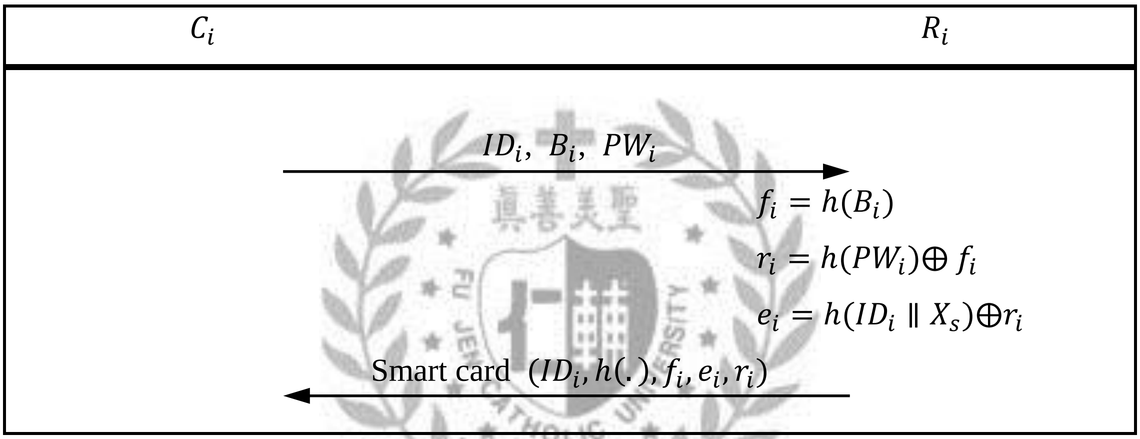

C_i
R_i
$ID_i,\ B_i,\ PW_i$
$f_i = h(B_i)$
$r_i = h(PW_i) \oplus f_i$
$e_i = h(ID_i \parallel X_s) \oplus r_i$
Smart card $(ID_i, h(.), f_i, e_i, r_i)$

Figure 2.3.1 Registration phase of Das's protocol

2.3.2 Login Phase

In this phase, when a user C_i wants to login to the server S_i, as shown in Figure 2.3.2, he/she needs to perform the following steps:

(1) C_i inserts his/her smart card into the card reader of a terminal and offers his/her personal biometric template B_i on a specific device to verify the biometric.

(2) Verifies whether B_i matches with the template stored in the system or not.

(3) If the above verification doesn't hold, then C_i doesn't pass the biometric verification. As a result, the remote user authentication is terminated. Otherwise, if

the above verification holds, C_i passes the biometric verification and inputs his/her password PW_i to perform the following step 4.

(4) The smart card computes $r_i' = h(PW_i) \oplus f_i$. The client terminates the session if $r_i' \neq r_i$.

(5) If $r_i' = r_i$, the smart card computes the followings:

$M_1 = e_i \oplus r_i'$, which is equal to $h(ID_i \parallel X_s)$,

$M_2 = M_1 \oplus R_c$, which is equal to $h(ID_i \parallel X_s) \oplus R_c$ and

$M_3 = h(R_c)$, where R_c is a random number generated by the user.

(6) Finally, C_i sends the message $\langle ID_i, M_2, M_3 \rangle$ to the remote server S_i.

C_i	S_i
Inserts the smart card and inputs B_i Verifies whether B_i matches with the template stored in the system or not Inputs PW_i $r_i' = h(PW_i) \oplus f_i$ Checks $r_i' ?= r_i$ $M_1 = e_i \oplus r_i'$ $M_2 = M_1 \oplus R_c$ $M_3 = h(R_c)$ $\xrightarrow{\langle ID_i, M_2, M_3 \rangle}$	

Figure 2.3.2 Login phase of Das's protocol

2.3.3 Authentication Phase

After receiving the login request messeges $\langle ID_i, M_2, M_3 \rangle$, the server S_i performs the following steps, as shown in Figure 2.3.3.

(1) S_i first checks the format of C_i's ID_i.

(2) If the format is valid, S_i then computes the following:

$M_4 = h(ID_i \parallel X_s)$ using the secret value maintained by the server.

$M_5 = M_2 \oplus M_4$, which needs to be R_c.

S_i verifies $h(M_5)? = M_3$. If it doesn't hold, S_i rejects C_i's login request. Otherwise, if the verification is successful, S_i computes the followings:

$M_6 = M_4 \oplus R_s = h(ID_i \parallel X_s) \oplus R_s$,

$M_7 = h(M_2 \parallel M_5) = h\big((h(ID_i \parallel X_s) \oplus R_c) \parallel R_c\big)$,

$M_8 = h(R_s)$.

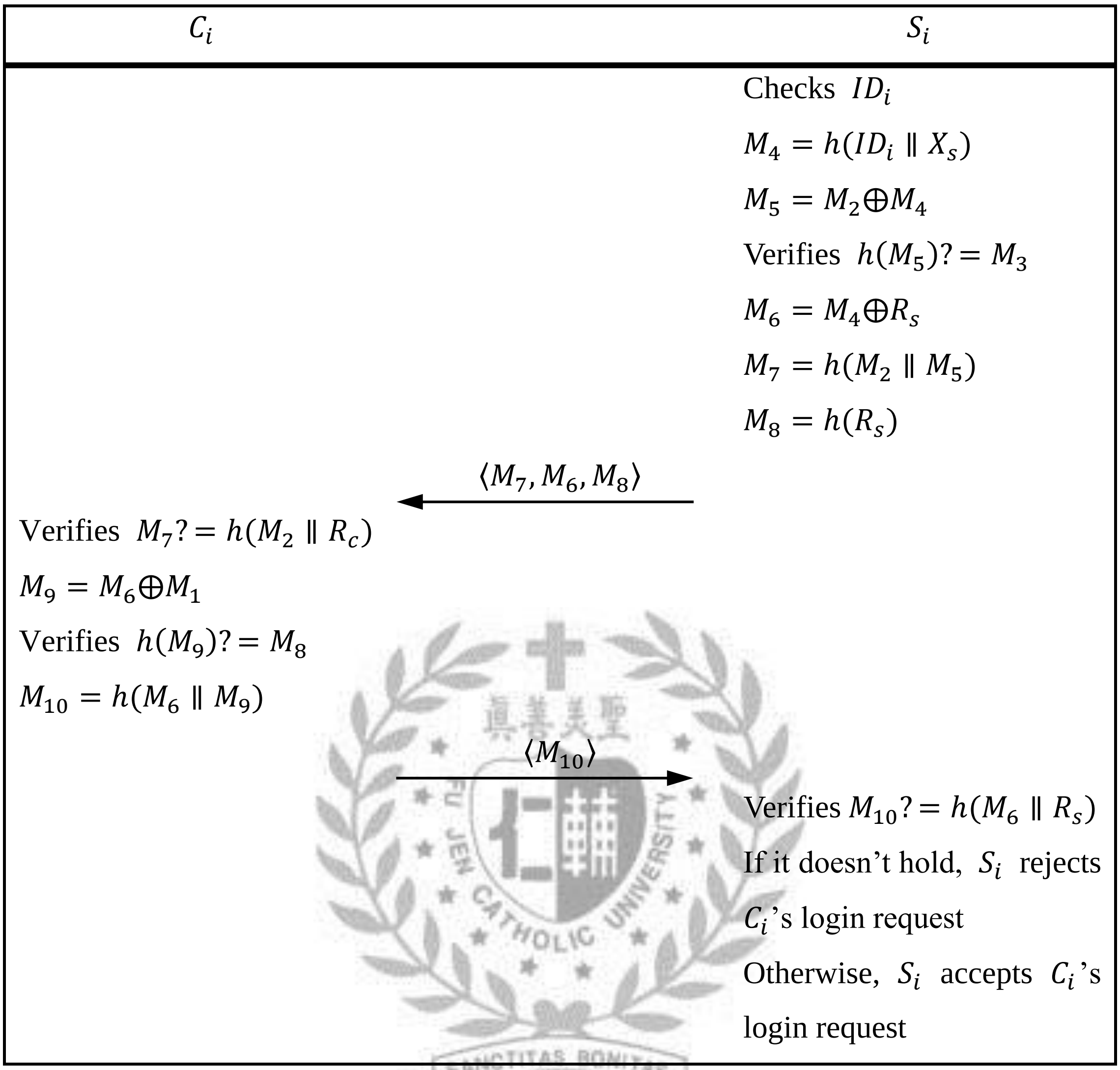
(3) S_i sends the messages $\langle M_7, M_6, M_8 \rangle$ to C_i.

(4) After receiving the messages $\langle M_7, M_6, M_8 \rangle$, C_i verifies $M_7? = h(M_2 \parallel R_c)$. Thus, C_i terminates the session if the verification doesn't pass. Otherwise, C_i computes $M_9 = M_6 \oplus M_1$ and verifies $h(M_9)? = M_8$. If $h(M_9) \neq M_8$, C_i terminates the session. Otherwise, C_i computes $M_{10} = h(M_6 \parallel M_9) = h\big((h(ID_i \parallel X_s) \oplus R_s) \parallel R_s\big)$ and sends the message M_{10} to the server S_i.

(5) After receiving C_i's message, S_i verifies $M_{10}? = h(M_6 \parallel R_s)$.

(6) S_i rejects C_i's login request if the above mentioned doesn't hold.

(7) Thus, S_i accepts C_i's login request if the verification is successful.



C_i		S_i
		Checks ID_i $M_4 = h(ID_i \parallel X_s)$ $M_5 = M_2 \oplus M_4$ Verifies $h(M_5)? = M_3$ $M_6 = M_4 \oplus R_s$ $M_7 = h(M_2 \parallel M_5)$ $M_8 = h(R_s)$
	$\xleftarrow{\langle M_7, M_6, M_8 \rangle}$	
Verifies $M_7? = h(M_2 \parallel R_c)$ $M_9 = M_6 \oplus M_1$ Verifies $h(M_9)? = M_8$ $M_{10} = h(M_6 \parallel M_9)$		
	$\xrightarrow{\langle M_{10} \rangle}$	
		Verifies $M_{10}? = h(M_6 \parallel R_s)$ If it doesn't hold, S_i rejects C_i's login request Otherwise, S_i accepts C_i's login request

Figure 2.3.3 Authentication phase of Das's protocol

2.3.4 Password Change Phase

In this phase, the smart card always verifies the old entered password by the user before updating the new changed password. In order to change the password, the user performs the following steps:

(1) Inserts the smart card and offers B_i.

(2) Verifies whether B_i matches with the template stored in the system or not.

(3) If C_i passes the biometric verification, C_i enters his/her old password PW_i^{old} and new changed password PW_i^{new}.

(4) The smart card computes the following:

$r_i' = h(PW_i^{old}) \oplus f_i$.

If $r_i' \neq r_i$, it means that C_i enters the wrong old password and the password change phase is terminated. If $r_i' = r_i$, then the smart card computes

$r_i'' = h(PW_i^{new}) \oplus f_i$,

$e_i' = e_i \oplus r_i' = h(ID_i \parallel X_s)$,

$e_i'' = e_i' \oplus r_i''$.

(5) Finally, smart card replaces e_i and r_i with e_i'' and r_i'', respectively.

2.4 Weaknesses of Das's Protocol

In this section, we analyze the security of Das's protocol. We show that Das's protocol is vulnerable to privileged insider attack and the off-line password guessing attack. In addition, Das's protocol also cannot provide user anonymity. We now describe the details as follows.

2.4.1 Privileged Insider Attack

In a real environment, it is a common practice that many users use the same password to access different applications or servers for convenience in remembering long passwords and ease-of-use whenever required [18]. However, if a privileged insider of the registration center knows the password of the user C_i, he/she may try to impersonate C_i for accessing other servers where C_i could be a registered user. In Das's protocol, the user C_i sends his/her real identity ID_i and password PW_i to the registration center R_i

directly in the registration phase. Hence, the privileged insider could get C_i's password and use it to impersonate C_i for accessing different applications or servers. Consequently, Das's protocol is vulnerable to the privileged insider attack.

2.4.2 Off-line Password Guessing Attack

Kocher et al. [29] and Messerges et al. [60] have pointed out all the information in smart cards could be extracted by the side channel attack. We assume that an adversary has stolen user C_j's smart card and extracted the information $(ID_j, h(.), f_j, e_j, r_j)$ of the smart card in Das's protocol. Using the extracted f_j and r_j, the adversary could find the password PW_j of user C_j through the following steps.

(1) The adversary uses f_j and r_j to compute $h(PW_j) = f_j \oplus r_j$.

(2) Then, the adversary chooses a password $PW_j{}'$ and verifies $h(PW_j{}')? = h(PW_j)$.

(3) If $h(PW_j{}') = h(PW_j)$, the guess was correct. Otherwise, the adversary can make another guess and repeat the process.

As mentioned above, we show that an adversary can get the password of user C_j and use it to impersonate C_i for accessing different applications or servers. Hence, Das's protocol is vulnerable to off-line password guessing attack.

2.4.3 Inability of Providing User Anonymity

In Das's protocol, the user C_i sends his/her real identity ID_i to the server S_i directly in the login phase. All of other users also send their real identity to the server S_i directly in the login phase. Hence, an adversary can get the real identity of any user by intercepting the messages $\{ID_i, M_2, M_3\}$ transmitted between the user and the server. Therefore, Das's protocol cannot provide user anonymity.

2.5 Our improved Biometric-based Protocol

In this section, we present our improved protocol using extended chaotic maps. The notations used in this section are summarized in Table 2.5.1.

Table 2.5.1 The notations used in this section

Notation	***Definition***
C_i	client
R_i	trusted registration center
S_i	server
PW_i	password shared between C_i and S_i
ID_i	identity of the user C_i
B_i	biometric template of the user C_i
p	a large prime number that $p = 2^{130} + 7$
X_s	a random integer chosen by the registration center
s	a random number chosen by the registration center
$SPUB$	the public key of R_i, where $SPUB \equiv T_{X_s}(s) \bmod p$
R_c, R_s	two random integers
t_i	the time-stamp
$h(\cdot)$	a secure one-way hash function
$\|$	the concatenation operation
$\oplus$	the exclusive-or (XOR) operation

In the beginning, the registration center R_i selects a random number s, a random integer X_s, and computes $SPUB \equiv T_{X_s}(s) \bmod p$. The registration center R_i keeps the master secret key X_s secretly. There are four phases in our protocol including registration phase, login phase, authentication phase, and password change phase. From now, the detailed steps of these phases are described in the following subsections.

2.5.1 Registration Phase

When the remote user C_i wants to register and become a new legal user in the system, as shown in Figure 2.5.1, he/she needs to perform the following steps:

(1) The user offers his/her password PW_i, the identity ID_i, generates a random number N, and also inputs his/her personal biometric B_i on a specific device and computes $f_i = h(B_i)$. C_i then sends $\{ID_i,\ f_i = h(B_i),\ h(PW_i \parallel B_i \parallel N)\}$ to the registration center R_i via secure channel.

(2) The registration center R_i computes the following

$P_i = h(ID_i \parallel X_s)$,

$r_i = h(PW_i \parallel B_i \parallel N) \oplus f_i$,

$e_i = P_i \oplus r_i$.

R_i embedded $(ID_i, h(.), e_i, s, SPUB, p)$ in the user's smart card and sends the card to the user C_i via a secure channel.

(3) After receiving the smart card, C_i computes $BPW = B_i \oplus h(PW_i)$ and inserts the random number N and BPW into the smart card and finishes the registration.

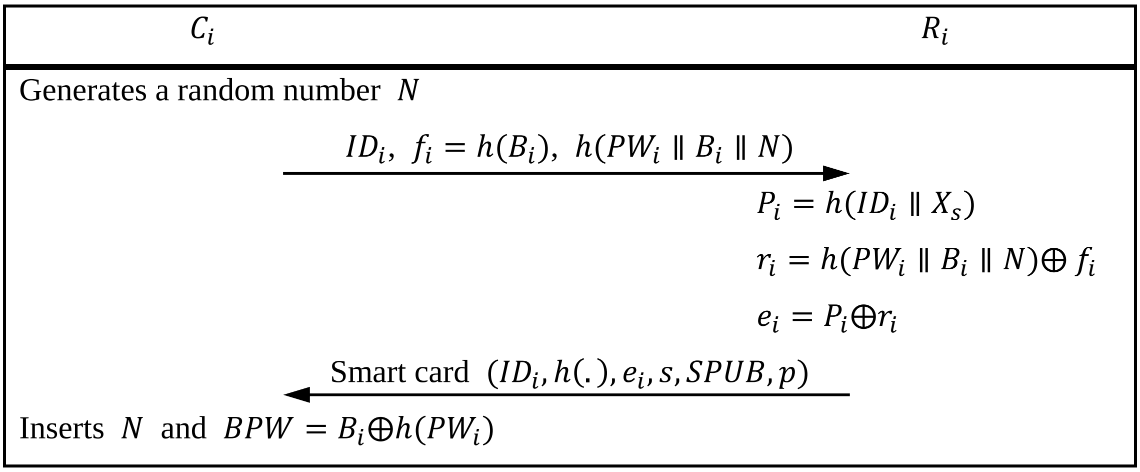

C_i
R_i
Generates a random number N
$ID_i,\ f_i = h(B_i),\ h(PW_i \parallel B_i \parallel N)$
$P_i = h(ID_i \parallel X_s)$
$r_i = h(PW_i \parallel B_i \parallel N) \oplus f_i$
$e_i = P_i \oplus r_i$
Smart card $(ID_i, h(.), e_i, s, SPUB, p)$
Inserts N and $BPW = B_i \oplus h(PW_i)$

Figure 2.5.1 Registration phase of our improved biometric-based protocol

2.5.2 Login Phase

In this phase, when a legal user C_i wants to access the server S_i, as shown in Figure 2.5.2, he/she needs to perform the following steps:

(1) C_i inserts his/her smart card into the card reader and offers both his/her personal biometric template B_i and password PW_i on a specific device.

(2) The smart card computes $B_i' = BPW \oplus h(PW_i)$ and verifies $B_i ?= B_i'$. If $B_i \neq B_i'$, the smart card rejects the request.

(3) The smart card generates a random integer R_c and computes

$f_i = h(B_i)$,

$r_i' = h(PW_i \parallel B_i \parallel N) \oplus f_i$,

$P_i' = e_i \oplus r_i'$,

$M_1 \equiv T_{R_c}(s) \bmod p$,

$M_2 \equiv T_{R_c}(SPUB) \bmod p$,

$NID_i = ID_i \oplus h(M_1 \parallel M_2)$,

$\alpha = h(ID_i \parallel NID_i \parallel P_i' \parallel M_1 \parallel M_2 \parallel t_1)$,

t_1 is the time-stamp which generated by the user C_i.

(4) The user C_i sends $\{NID_i,\ M_1,\ \alpha,\ t_1\}$ to S_i.

C_i	S_i
Inserts the smart card and inputs $PW_i,\ B_i$	
$B_i{}' = BPW \oplus h(PW_i)$	
Verifies $B_i ?= B_i{}'$	
Generates R_c	
$f_i = h(B_i)$	
$r_i{}' = h(PW_i \parallel B_i \parallel N) \oplus f_i$	
$P_i{}' = e_i \oplus r_i{}'$	
$M_1 \equiv T_{R_c}(s) \bmod p$	
$M_2 \equiv T_{R_c}(SPUB) \bmod p$	
$NID_i = ID_i \oplus h(M_1 \parallel M_2)$	
$\alpha = h(ID_i \parallel NID_i \parallel P_i{}' \parallel M_1 \parallel M_2 \parallel t_1)$	
$\xrightarrow{NID_i,\ M_1,\ \alpha,\ t_1}$	

Figure 2.5.2 Login phase of our improved biometric-based protocol

2.5.3 Authentication Phase

After receiving the login request messages, the server S_i performs the following steps to access mutual authentication, as shown in Figure 2.5.3.

(1) Upon receiving $\{NID_i,\ M_1,\ \alpha,\ t_1\}$, S_i first checks the validity of t_1 by checking whether the equation $t' - t_1 > \Delta t$ holds or not, where the t' is the time when the server receives the messages from C_i and Δt denotes the predetermined legal time interval of transmission delay. If the equation holds, S_i rejects C_i.

(2) S_i computes $M_2' \equiv T_{X_s}(M_1) \bmod p$, $ID_i{}' = NID_i \oplus h(M_1 \parallel M_2{}')$ and checks the validity of $ID_i{}'$.

(3) S_i computes $P_i{}'' = h(ID_i{}' \parallel X_s)$ and $\alpha' = h(ID_i{}' \parallel NID_i \parallel P_i{}'' \parallel M_1 \parallel M_2{}' \parallel t_1)$.

(4) Then S_i verifies whether α' equals to α. If $\alpha' \neq \alpha$, S_i stops the session.

(5) If $\alpha' = \alpha$, S_i randomly chooses an integer R_s and computes $M_3 \equiv T_{R_s}(s) \bmod p$ and $\beta = h(ID_i{}' \parallel P_i{}'' \parallel M_2{}' \parallel M_3 \parallel t_2)$. Then, S_i sends $\{M_3, \beta, t_2\}$ to C_i.

(6) After receiving $\{M_3, \beta, t_2\}$, C_i first checks the validity of t_2 by checking whether the equation $t' - t_2 > \Delta t$ holds. If the equation holds, C_i rejects S_i.

(7) C_i computes $\beta' = h(ID_i \parallel P_i{}' \parallel M_2 \parallel M_3 \parallel t_2)$ and verifies whether $\beta'? = \beta$. If it doesn't equal, C_i stops the session. Otherwise, C_i computes $M_4 \equiv T_{R_c}(M_3) \equiv T_{R_cR_s}(s) \bmod p$ and $\gamma = h(ID_i \parallel P_i{}' \parallel M_2 \parallel M_4 \parallel t_3)$. C_i then sends $\{\gamma, t_3\}$ to S_i.

(8) Upon receiving $\{\gamma, t_3\}$, S_i first checks the validity of t_3 by checking whether the equation $t' - t_3 > \Delta t$ holds. If the equation holds, S_i rejects C_i. Otherwise, S_i computes $M_4' \equiv T_{R_s}(M_1) \equiv T_{R_cR_s}(s) \bmod p$ and $\gamma' = h(ID_i{}' \parallel P_i{}'' \parallel M_2{}' \parallel M_4{}' \parallel t_3)$ and checks whether $\gamma'? = \gamma$.

(9) If it holds, S_i accepts C_i's login request and the verification is successful. Then both C_i and S_i can use the session key M_4 and M_4' to communicate with each other by using a symmetric cryptosystem.

Since $SPUB \equiv T_{X_s}(s) \bmod p$, $M_1 \equiv T_{R_c}(s) \bmod p$, $M_2 \equiv T_{R_c}(SPUB) \bmod p$, and $M_3 \equiv T_{R_s}(s) \bmod p$, so we can derive

$$M_2' \equiv T_{X_s}(M_1) \equiv T_{X_s}\left(T_{R_c}(s)\right) \equiv T_{R_c}\left(T_{X_s}(s)\right) \equiv T_{R_c}(SPUB) \equiv M_2 \bmod p$$

and

$$M_4' \equiv T_{R_c}(M_3) \equiv T_{R_c}\left(T_{R_s}(s)\right) \equiv T_{R_s}\left(T_{R_c}(s)\right) \equiv T_{R_s}(M_1) \equiv M_4 \bmod p.$$

Therefore, the correctness of the protocol is proved.

C_i		S_i
		Checks $t' - t_1 > \Delta t$ $M_2' \equiv T_{X_s}(M_1) \bmod p$ $ID_i' = NID_i \oplus h(M_1 \parallel M_2')$ Checks ID_i' $P_i'' = h(ID_i' \parallel X_s)$ $\alpha' = h(ID_i' \parallel NID_i \parallel P_i'' \parallel M_1 \parallel M_2' \parallel t_1)$ Verifies $\alpha' ?= \alpha$ Generates R_s $M_3 \equiv T_{R_s}(s) \bmod p$ $\beta = h(ID_i' \parallel P_i'' \parallel M_2' \parallel M_3 \parallel t_2)$
	$\xleftarrow{M_3,\ \beta,\ t_2}$	
Checks $t' - t_2 > \Delta t$ $\beta' = h(ID_i \parallel P_i' \parallel M_2 \parallel M_3 \parallel t_2)$ Verifies $\beta' ?= \beta$ $M_4 \equiv T_{R_c}(M_3) \equiv T_{R_cR_s}(s) \bmod p$ $\gamma = h(ID_i \parallel P_i' \parallel M_2 \parallel M_4 \parallel t_3)$		
	$\xrightarrow{\gamma,\ t_3}$	
		Checks $t' - t_3 > \Delta t$ $M_4' \equiv T_{R_s}(M_1) \equiv T_{R_cR_s}(s) \bmod p$ $\gamma' = h(ID_i' \parallel P_i'' \parallel M_2' \parallel M_4' \parallel t_3)$ Verifies $\gamma' ?= \gamma$

Figure 2.5.3 Authentication phase of our improved biometric-based protocol

2.5.4 Password Change Phase

In this phase, the smart card always verifies the old entered password by the user before updating the new changed password. In order to change the password, the user C_i performs the following steps:

(1) Inserts the smart card and offers both the biometric template B_i and old password PW_i.

(2) The smart card computes $B_i{}' = BPW \oplus h(PW_i)$ and verifies $B_i ?= B_i{}'$. If $B_i \neq B_i{}'$, it means that C_i enters the wrong old password or the wrong biometric template. Then, the smart card rejects the request.

(3) If C_i passes the biometric verification, C_i enters his/her new password PW_i^{new}.

(4) The smart card computes the following:

$f_i = h(B_i)$,

$r_i{}' = h(PW_i \parallel B_i \parallel N) \oplus f_i$,

$r_i{}'' = h(PW_i^{new} \parallel B_i \parallel N) \oplus f_i$,

$P_i{}' = e_i \oplus r_i{}'$,

$e_i{}' = P_i{}' \oplus r_i{}''$.

(5) Finally, replaces the e_i with $e_i{}'$ on the smart card.

2.6 Analysis of our improved Biometric-based Protocol

In this section, we analyze the security and performance of our improved protocol and show it could overcome the security weaknesses of Das's protocol. Then, we will describe the details as following.

2.6.1 Security Analysis

Here, we describe several security analyses of our improved protocol.

- Privileged Insider Attack

In the registration phase of our improved protocol, the remote user C_i sends $h(PW_i \parallel B_i \parallel N)$ to the registration center R_i. The privileged insider cannot derive the password PW_i without B_i and N. Therefore, our improved protocol can resist the privileged insider attack.

- Replay Attack

The attacker may intercept the communication messages from C_i and replay them to the server S_i in next run. However, the attacker cannot pass the verification with the incorrect timestamps. Hence, our improved protocol is secure against the replay attack by using the timestamps t_1, t_2, and t_3.

- Off-line Password Guessing Attack

The attacker may intercept the messages $\{NID_i,\ M_1,\ \alpha,\ t_1\}$ and $\{M_3,\ \beta,\ t_2\}$. The attacker may also get e_i stored in the smart card. Then he/she could try to guess the password $PW_i{}'$. But the attacker cannot verify the correctness of the password $PW_i{}'$ since he/she does not know the elements r_i, f_i, B_i and P_i. If the attacker wants to derive the random integers R_c and R_s, he/she will also face the DHP. Therefore, our improved protocol can resist the off-line password guessing attack.

- User Anonymity

The attacker may eavesdrop on the communication between user C_i and server S_i, and try to track the user's real identity to find some information of the user. In our improved protocol, the real identity ID_i is protected by $M_2 \equiv M_2' \equiv T_{X_s}\left(T_{R_c}(s)\right) \bmod p$ from $PUB \equiv T_{X_s}(s) \bmod p$ and $M_1 \equiv T_{R_c}(s) \bmod p$. In order to compute M_2, the attacker will face the DHP. Hence, our improved protocol can provide the user anonymity.

- Mutual Authentication

  Our protocol can achieve mutual authentication between user C_i and server S_i. In the authentication phase of our improved protocol, server S_i has to verify the validity of α and γ in order to authenticate C_i. The user C_i's smart card also has to verify the validity of β in order to authenticate S_i. If there is an attacker who wants to forge the messages, he/she will face the DLP and the DHP. Therefore, both the user and the server can authenticate with each other, and mutual authentication between them is achieved.

- Stolen-verifier Attack

  The stolen-verifier attack means that an attacker steals the security-sensitive verification table from the server and uses it to masquerade as a legitimate user in the authentication phase. The server in our improved protocol does not need to maintain any security-sensitive verification table. Hence, our improved protocol can resist the stolen-verifier attack.

- Lost Smart Card

  Assume that an attacker can extract all the information from the smart card by the side channel attack [29, 60]. The attacker may try to derive the password from the information, but the password is protected by the elements r_i, f_i, B_i and P_i that the attacker does not know them. Besides, the attacker also cannot pass the biometric verification without the user's biometric template B_i. Therefore, our improved protocol is secure against the smart card loss problem.

2.6.2 Performance Analysis

Here, we discuss the performance of our improved protocol. We compare the security properties of our improved protocol with Tseng et al.'s protocol [71], Lee et al.'s protocol [35], He et al.'s protocol [16], and Das's protocol [9] in Table 2.6.1. We also define some notations as follows:

- T_X: Time for performing an XOR operation
- T_H: Time for performing a one-way hash function
- T_E: Time for performing a symmetric encryption operation
- T_D: Time for performing a symmetric decryption operation
- T_C: Time for performing a Chebyshev chaotic map operation

In Table 2.6.1, we can see that our improved protocol is more secure than other protocols. We also compare the performance of our improved protocol with other protocols in Table 2.6.2. The costs of our improved protocol are slightly higher than Das's protocol. However, Das's protocol is vulnerable to the privileged insider attack, the off-line password guessing attack, and also cannot provide user anonymity. As a result, our improved protocol can overcome the weaknesses in Das's protocol and is more secure than his protocol.

Table 2.6.1 Comparison of security properties

	Tseng et al.'s protocol	Das's protocol	Lee et al.'s protocol	He et al.'s protocol	Our protocol
Privileged attack	✗	✗	✗	✗	✓
Replay attack	✓	✓	✓	✓	✓
Off-line guessing attack	✓	✗	✓	✓	✓
User anonymity	✗	✗	✗	✓	✓
Mutual authentication	✗	✓	✓	✓	✓

Table 2.6.2 Comparison of performance

	Client	Server
Tseng et al.'s protocol	$2T_X + 5T_H + 1T_E + 1T_D + 1T_C$	$1T_X + 3T_H + 1T_E + 1T_D + 2T_C$
Das's protocol	$4T_X + 5T_H$	$4T_X + 8T_H$
Lee et al.'s protocol	$6T_X + 6T_H + 2T_C$	$6T_X + 6T_H + 2T_C$
He et al.'s protocol	$2T_X + 5T_H + 3T_C$	$2T_X + 5T_H + 3T_C$
Our protocol	$5T_X + 10T_H + 3T_C$	$3T_X + 7T_H + 3T_C$

Chapter 3 A Three-party Password-based Authenticated Key Exchange Protocol with User Anonymity Using Extended Chaotic Maps

In this thesis, we propose a protocol utilizing three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps, which is more efficient and secure than previously proposed protocols. In order to enhance the efficiency and security, we use the extended chaotic maps to encrypt and decrypt the information transmitted by the user or the server. In addition, the proposed protocol provides user anonymity to guarantee the identity of users, which is transmitted in the insecure public network.

3.1 Preliminaries

In order to guarantee the security of secret keys which are exchanged over the insecure public network, there are many related protocols [6, 7, 40, 41, 49, 61] which have been proposed by researchers, such as Password-Authenticated Key Exchange (PAKE) protocol. PAKE protocol allows two parties to keep one identical memorable password to agree on a common session key over the insecure public network [16, 53, 54, 69]. Generally, password-based authentication can resist both the brute force and the dictionary attacks if users choose strong passwords to provide enough entropy. Nevertheless, password-based authentication has one intrinsic problem: users are not adept in memorizing text strings. Therefore, it is not easy to protect the password information against various attacks since most users would select memorable passwords even if they know the passwords might be unsafe. According to the protocol proposed by Lin et al. [52], we can divide the attacks into the following classes:

- Off-line dictionary attacks: The adversary first guesses a password and then verifies its guess in an off-line mode only by using the eavesdropped information. No participation of the honest client or the server is required, so these attacks cannot be noticed.
- Undetectable on-line dictionary attacks: The adversary attempts to verify a password guess in an on-line transaction. Nevertheless, a failed guess cannot be detected by the honest client or by the server, since one of them is not able to distinguish a malicious request from an honest one.
- Detectable on-line dictionary attacks: Similar to above, the adversary tries to use a guessed password in an on-line transaction. The adversary verifies the correctness of its guess by using the response from the honest client or the server. But a failed guess can be detected by the honest client or the server.

Among these attacks, both off-line and undetectable on-line dictionary attacks can cause serious consequences against password-based authentication protocol. Consequently, it is a crucial consideration to design a secure password-based authentication protocol which can resist the mentioned above attacks.

In 1992, Bellovin and Merritt [2] proposed the first PAKE protocol. After a decade, many related protocols, such as both the two-party PAKE [6, 7, 49] and the three-party PAKE [32, 33, 40, 41, 61, 81] have been proposed. However, Hassan and Abdullah [15] pointed out that two-party PAKE protocols are not suitable in the large peer-to-peer architecture. Also, some of the three-party PAKE protocols are not secure or efficient enough to be used in practice. Recently, Abdalla et al. [1] and Lu et al. [56] proposed two efficient three-party password-based key exchange protocols in 2005 and 2007 respectively. Unfortunately, both of their protocols were still vulnerable to undetectable on-line dictionary attacks or off-line dictionary attacks. In 2009, Deng et al. [10] proposed a three party password-based key exchange protocol and declared that their protocol was secure under the universal composable framework (UC-SECURE). However, Yuan et al. [85] pointed out that Deng et al.'s protocol is insecure against offline dictionary attack by any other client. In 2011, Yoon and Yoo proposed a protocol [83] and pointed out that Huang's protocol [19] could not resist undetectable on-line dictionary attacks and key-compromise impersonation attack. Subsequently, Yoon and Yoo also proposed another protocol [82] and showed that Lou and Huang's protocol [55] was vulnerable to off-line password guessing attacks by an attacker. After that, Wu et al. [78] also found the security weaknesses of Huang's protocol [19] and proposed a three-party password-based authenticated key exchange protocol to remedy the security flaws in Huang's protocol. Nevertheless, Wu et al.'s protocol had many exponential computations, which required the highest computational complexity and could not provide user anonymity.

In order to enhance the efficiency and security, we propose a three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps. The proposed protocol uses the extended chaotic maps both to encrypt and to decrypt the information transmitted by the user or the server. It can also provide mutual authentication between user and server, and user anonymity to guarantee the identity of users which is transmitted in the insecure public network. The security and performance analysis show that the proposed protocol has low computation and communication cost and can also resist against various attacks.

3.2 Our Proposed Protocol

In this section, the proposed protocol with user anonymity using extended chaotic maps is described in detail, which is based on Wu et al.'s protocol [78]. The notations used in this Chapter are summarized in Table 3.2.1.

Table 3.2.1 The notations used in Chapter 3

Notation	***Definition***
A, B	two identity of clients (users)
TS	a trusty server
PW_A, PW_B	the password shared between user A (resp. B) and server TS
p	a large prime number that $p = 2^{130} + 7$
s	a random integer chosen by TS
r	a random number chosen by TS
Q	the public key of TS, where $Q \equiv T_s(r) \bmod p$
SK	the session key used between user A and B
x, y	two random integers
t_1	the time-stamp
$h(\cdot)$	a secure hash function
$\|$	the concatenation operation
$\oplus$	the exclusive-or (XOR) operation

In the beginning, the remote server TS selects a random number r, a random integer s, and computes its public key $Q \equiv T_s(r) \bmod p$. The remote server TS keeps its private key s secretly. In our protocol, we assume the two users A and B have already established the common secret key share passwords PW_A, PW_B with the remote server TS, respectively. The remote server TS distributes the public parameters $(Q, r, h(\cdot), p)$ to all parties in the network. The simplified description of the proposed protocol is shown in Figure 3.2.1. From this point, the details of the proposed protocol are described in the following steps:

(1) User A chooses a random integer x and computes the followings

$R_A \equiv T_x(r) \bmod p$,

$T_A \equiv T_x(Q) \bmod p$,

$AID_i = A \oplus h(T_A)$,

$\tau_{A,S} = h(A \parallel B \parallel TS \parallel AID_i \parallel PW_A \parallel T_A \parallel t_1)$.

User A sends $(AID_i, R_A, \tau_{A,S}, t_1)$ to user B.

(2) After receiving $(AID_i, R_A, \tau_{A,S}, t_1)$, user B chooses a random integer y and computes the followings

$R_B \equiv T_y(r) \bmod p$,

$T_B \equiv T_y(Q) \bmod p$,

$BID_i = B \oplus h(T_B)$,

$\tau_{B,S} = h(B \parallel TS \parallel BID_i \parallel PW_B \parallel T_B)$.

Then user B sends $(AID_i, R_A, \tau_{A,S}, BID_i, R_B, \tau_{B,S}, t_1)$ to the remote server TS.

(3) Upon receiving $(AID_i, R_A, \tau_{A,S}, BID_i, R_B, \tau_{B,S}, t_1)$, the server TS first checks the validity of t_1 by checking whether the equation $t' - t_1 > \Delta t$ holds, where the t' is the time when the server receives the messages from B. Δt denotes the predetermined legal time interval of transmission delay. If the equation does not hold, then the server TS calculates $T_A{}' \equiv T_s(R_A) \bmod p$, $T_B{}' \equiv T_s(R_B) \bmod p$, $A' = AID_i \oplus h(T_A{}')$, and $B' = BID_i \oplus h(T_B{}')$ and uses them to check $\tau_{A,S}$ and $\tau_{B,S}$ respectively. If the values are invalid, TS terminates the protocol. Otherwise, TS computes $\tau_{S,A} = h(A' \parallel B' \parallel TS \parallel PW_A \parallel T_A{}')$, $\tau_{S,B} = h(A' \parallel B' \parallel TS \parallel PW_B \parallel T_B{}')$, and $AID_j = A' \oplus h(T_B{}')$ and then sends $(\tau_{S,A}, \tau_{S,B}, AID_j)$ to user B.

(4) After receiving $(\tau_{S,A}, \tau_{S,B}, AID_j)$, user B first computes $A'' = AID_j \oplus h(T_B)$ and checks the validity of $\tau_{S,B}$ using T_B. If the value is invalid, B terminates the protocol. Otherwise, both server TS and user A are authenticated by user B and B computes the common session key $SK \equiv T_y(R_A) \bmod p$ and $S_{BA} = h(SK \parallel A'' \parallel B)$. Finally, B sends $(R_B, \tau_{S,A}, S_{BA})$ to user A.

(5) Upon receiving $(R_B, \tau_{S,A}, S_{BA})$, user A first checks the validity of $\tau_{S,A}$ using T_A. If the value is invalid, A terminates the protocol. Otherwise, user A computes the common session key $SK \equiv T_x(R_B) \bmod p$ and checks the validity of $S_{BA} = h(SK \parallel A \parallel B)$. If it does not hold, A terminates the protocol. Otherwise, both server TS and user B are authenticated by user A and the common session key SK is agreed upon. Then, user A computes $S_{AB} = h(SK \parallel A \parallel R_B)$ and sends it to B.

(6) After receiving (S_{AB}), user B checks the validity of $S_{AB} = h(SK \parallel A'' \parallel R_B)$. If it does not hold, B terminates the protocol. Otherwise, both user A and user B can use the common session key SK for secure communication. The common session key SK is only used for one session.

User A	User B	Trusted server TS
PW_A	PW_B	$(s, Q \equiv T_s(r) \bmod p)$
Generate x		
$R_A \equiv T_x(r) \bmod p$		
$T_A \equiv T_x(Q) \bmod p$		
$AID_i = A \oplus h(T_A)$		
$\tau_{A,S} = h(A \parallel B \parallel TS \parallel AID_i \parallel PW_A \parallel T_A \parallel t_1)$		
$\xrightarrow{(AID_i, R_A, \tau_{A,S}, t_1)}$		
	Generate y	
	$R_B \equiv T_y(r) \bmod p$	
	$T_B \equiv T_y(Q) \bmod p$	
	$BID_i = B \oplus h(T_B)$	
	$\tau_{B,S} = h(B \parallel TS \parallel BID_i \parallel PW_B \parallel T_B)$	
	$\xrightarrow{(AID_i, R_A, \tau_{A,S}, BID_i, R_B, \tau_{B,S}, t_1)}$	
		Check $t' - t_1 > \Delta t$
		$T_A{}' \equiv T_s(R_A) \bmod p$
		$T_B{}' \equiv T_s(R_B) \bmod p$
		$A' = AID_i \oplus h(T_A{}')$
		$B' = BID_i \oplus h(T_B{}')$
		Check $\tau_{A,S}$ and $\tau_{B,S}$
		$\tau_{S,A} = h(A' \parallel B' \parallel TS \parallel PW_A \parallel T_A{}')$
		$\tau_{S,B} = h(A' \parallel B' \parallel TS \parallel PW_B \parallel T_B{}')$
		$AID_j = A' \oplus h(T_B{}')$
		$\xleftarrow{(\tau_{S,A}, \tau_{S,B}, AID_j)}$
	$A'' = AID_j \oplus h(T_B)$	
	Check $\tau_{S,B}$	
	$SK \equiv T_y(R_A) \bmod p$	
	$S_{BA} = h(SK \parallel A'' \parallel B)$	
	$\xleftarrow{(R_B, \tau_{S,A}, S_{BA})}$	
Check $\tau_{S,A}$		
$SK \equiv T_x(R_B) \bmod p$		
Verify: $S_{BA} = h(SK \parallel A \parallel B)$		
$S_{AB} = h(SK \parallel A \parallel R_B)$		
$\xrightarrow{(S_{AB})}$		
	Verify: $S_{AB} = h(SK \parallel A'' \parallel R_B)$	

Figure 3.2.1 The proposed three-party password-based authenticated key exchange protocol

3.3 Security Analysis and Comparison

In this section, we analyze the security and performance of our protocol and show it could resist against various attacks. Here, we describe several security analyses in the proposed protocol.

3.3.1 Off-line Dictionary Attacks

The attacker may intercept the messages $(AID_i, R_A, \tau_{A,S}, t_1)$ or $(AID_i, R_A, \tau_{A,S}, BID_i, R_B, \tau_{B,S}, t_1)$ and try to guess the password from the element $\tau_{A,S}$ or $\tau_{B,S}$. However, the attacker cannot successfully verify the password without knowing T_A or T_B, which are generated by user A and B respectively based on the difficulty of the DLP problem. Hence, the proposed protocol is secure against the off-line dictionary attacks.

3.3.2 Undetectable On-line Dictionary Attacks

The attacker may intercept the messages $(AID_i, R_A, \tau_{A,S}, t_1)$ or $(AID_i, R_A, \tau_{A,S}, BID_i, R_B, \tau_{B,S}, t_1)$ and try to impersonate a legal user. But the attacker cannot send a new valid message $(AID_i, R_A, \tau_{A,S}, BID_i, R_B, \tau_{B,S}, t_1)$ to the trusted server unless he/she has guessed the correct password. Moreover, if the attacker tries to guess the password, he/she will face the DLP problem. Therefore, the proposed protocol can resist the undetectable on-line dictionary attacks.

3.3.3 Detectable On-line Dictionary Attacks

The attacker may intercept the messages $(AID_i, R_A, \tau_{A,S}, t_1)$ or $(AID_i, R_A, \tau_{A,S}, BID_i, R_B, \tau_{B,S}, t_1)$ and try to impersonate a legal user. But the attacker cannot send a new valid message $(AID_i, R_A, \tau_{A,S}, BID_i, R_B, \tau_{B,S}, t_1)$ to the trusted server unless he/she has guessed the correct password. Moreover, the server will check the correctness of $\tau_{A,S}$ and $\tau_{B,S}$. Hence, the attacker will be detected if he/she sends an invalid message to the server. In that case the proposed protocol is secure against the detectable on-line dictionary attacks.

3.3.4 Replay Attack

The attacker may intercept the messages from a user and replay them to the server in the next run. Nevertheless, the server could find the attack by checking the validity of the timestamp t_1. The attacker may also intercept the messages from the server and replay it to user. However, the users have generated the new random integers x and y. Then user A and B could find the attack by verifying the correctness of $\tau_{S,A}$ and $\tau_{S,B}$ respectively. Hence, the proposed protocol can resist the replay attack.

3.3.5 User Anonymity

The attacker may eavesdrop the communication between the user and the trusted server, and try to trace the user's real identity to find some security-sensitive information of the user. In the proposed protocol, the real identity of user A and B are protected by $AID_i = A \oplus h(T_A)$ and $BID_i = B \oplus h(T_B)$ respectively. In order to compute T_A and

T_B, the attacker will face the DLP problem. Hence, the proposed protocol can provide the user with a high degree of anonymity.

3.3.6 Mutual Authentication

The proposed protocol can achieve mutual authentication between the user and the server. In step 3 of the proposed protocol, the server TS must verify the validity of $\tau_{A,S}$ and $\tau_{B,S}$ in order to authenticate user A and B. User A and B also must verify the validity of $\tau_{S,A}$ and $\tau_{S,B}$ respectively in order to authenticate server TS. If there is an attacker who wants to forge messages, he/she will face not only the DLP but also the DHP problems. Therefore, as both the user and the trusted server can authenticate each other, the mutual authentication between them is achieved.

3.4 Performance Discussion and Comparison

In this section, we discuss the performance of the proposed protocol. We compare the security properties of the proposed protocol with Huang's protocol [19], Lou and Huang's protocol [55], Lee et al.'s protocol [34], and Wu et al.'s protocol [78] in Table 3.4.1.

In Table 3.4.1, we can see that the proposed protocol is more secure than other protocols. We also compare the performance of the proposed protocol with other protocols in Table 3.4.2. In Table 3.4.2, U denotes the user and S denotes the server. The computational complexity of modular exponential is higher than all other operations such as hash computation and Chebyshev chaotic maps, which can be done efficiently. The proposed protocol is more efficient than other protocols even if the costs of the proposed protocol are slightly higher than Lou and Huang's protocol. However, Lou and Huang's protocol is vulnerable to the off-line dictionary attacks and also cannot provide user anonymity. As shown in Table 3.4.1, none of the other protocols can provide user anonymity. Consequently, the proposed protocol is more efficient and secure than others since our protocol only uses hash operation and XOR operation and also can provide user anonymity.

Table 3.4.1 Comparison of security properties

	Huang's protocol	Lou and Huang's protocol	Lee et al.'s protocol	Wu et al.'s protocol	Our protocol
Off-line dictionary attacks	✓	✗	✓	✓	✓
Undetectable on-line dictionary attacks	✗	✓	✓	✓	✓
Replay attack	✗	✓	✓	✓	✓
User anonymity	✗	✗	✗	✗	✓
Mutual authentication	✗	✓	✓	✓	✓

Table 3.4.2 Comparison of performance

	Huang's protocol		Lou and Huang's protocol		Lee et al.'s protocol		Wu et al.'s protocol		Our protocol	
	U	S	U	S	U	S	U	S	U	S
Modular exponential	4	2	0	0	6	4	8	2	0	0
Hash/TDF operation	6	4	4	2	2	2	6	2	8	5
Chebyshev chaotic map operation	0	0	0	0	0	0	0	0	6	2
Random number	2	1	2	1	4	1	4	0	2	0
XOR operation	4	4	2	2	2	2	2	2	3	3
Round	5		5		5		5		5	

Chapter 4 An Extended Chaotic-maps-based Protocol with Key Agreement for Multi-server Environments

Due to the rapid development and growth of computer networks, there have been greater and greater demands for remote password authentication protocols. Recently, the focus has been on protocols for multi-server environments that run on smart cards. These protocols typically count on the nonce or timestamp to provide protection against the replay attack. Nevertheless, as Tsaur et al. pointed out, these protocols have some security issues such as disturbance in clock synchronization and vulnerability to the man-in-the-middle attack. In order to solve the mentioned problems, Tsaur et al. proposed a multi-server authentication protocol with key agreement in 2012, and they claimed that their protocol could effectively achieve password-authenticated key agreement while getting around the technical difficulty of implementing clock synchronization in multi-server environments. Unfortunately, we found that Tsaur et al.'s protocol still has the following weaknesses: (1) inability to resist privileged insider attack, (2) inability to resist known-plaintext attack, (3) inability to provide user anonymity, and (4) lack of perfect forward secrecy. To fix these secure flaws of Tsaur et al.'s protocol, we shall propose an improved multi-server authentication protocol with key agreement based on extended chaotic maps. We also offer formal proof of smooth execution of our improved authenticated key agreement protocol.

4.1 Preliminaries

As the electronics industry flourishes, devices of a tiny size with low power consumption such as the smart card have been gaining popularity. As a result, smart-card-related applications have formed a bigger and bigger market worldwide. On the other hand, in order to protect confidential information stored in servers from being accessed by any malicious party, password authentication protocols have been created and well received because of their simple implementation, easy operation, and low cost [46, 47, 59, 70]. Naturally, many researchers now have started to take advantage of the convenience and swiftness the smart card offers as they try to enhance the efficiency and security of password authentication protocols [17, 20, 67, 72]. Traditional password authentication protocols were mostly designed to be used in single server environments [5, 22, 24, 39]. However, with the rapid advancement and extensive implementation of multi-server systems in computer networks, traditional password authentication protocols have obviously fallen out of date [70]. Now, among the new password authentication protocols especially designed for multi-server environments, the Kerberos system [30] is one of the most well-known. Nevertheless, in the Kerberos system, members need to use strong cryptographic secrets for the authentication to work properly; in other words, the system is insecure against password guessing attacks if the user picks a weak password.

Recently, quite a number of convenient password authentication protocols have been proposed especially for the maintenance of system security in multi-server environments. In 2001, Li et al. [48] proposed a remote password authentication protocol for multi-server architecture using neural networks. The key feature of their system is that it can withstand the replay attack but does not need to maintain a verification table. Unfortunately, in 2003, Lin et al. [50] pointed out that Li et al.'s protocol spends too much time training neural networks, and so they proposed their own improved version of the remote user authentication protocol for multi-server architecture in order to enhance the efficiency of Li et al.'s protocol. In 2004, Juang [25] proposed an efficient remote password authentication protocol to be used in a multi-server environment and demonstrated that his protocol could satisfy all the requirements in seven important criteria. In 2008, Tsai [68] also proposed a multi-server authentication protocol based on the one-way hash function without using a verification table. Nevertheless, neither Juang's protocol nor Tsai's protocol can resist the man-in-the-middle attack. Most importantly, all of these protocols rely on the nonce or timestamp to keep the replay attack from working, but this will require the cost for implementing clock synchronization. For solving the above-mentioned problems, in 2012, Tsaur et al. [70] proposed a technique of self-verified timestamp where the timestamp is verified by the timestamp creator. They claimed that their protocol could effectively achieve password-authenticated key agreement and save the trouble of implementing clock synchronization in multi-server environments. However, we found that Tsaur et al.'s protocol still has the following security flaws: (1) it cannot resist the privileged insider attack; (2) it cannot resist the known-plaintext attack; (3) it is unable to provide user anonymity; (4) it does not provide perfect forward secrecy. To make Tsaur et al.'s protocol stronger, Li et al. [47] improved it into an extended multi-server-based user authentication and key agreement protocol with user anonymity. Unfortunately, Li et al.'s protocol still has the same secure flaws we pointed out. Therefore, we propose an improved multi-server authentication protocol with key agreement based on extended chaotic maps. Our new chaotic-maps-based protocol will not only allow users to anonymously communicate with the server but also provide mutual authentication between user and server.

4.2 Review of Tsaur et al.'s Protocol

In this section, we shall review and analyze Tsaur et al.'s protocol [70]. Tsaur et al.'s protocol has two phases to it: the registration phase and the log-in and session key agreement phase. The notations used throughout Tsaur et al.'s protocol are summarized in Table 4.2.1. Let x be the master secret key created and kept in secrecy by the registration center (RC). RC computes the secret key $w_j = h(x \parallel SID_j)$, which is to be shared between it and the jth server S_j, where SID_j is the jth server's identity and $h(\cdot)$ is a one-way and collision-free hash function with a fixed 160-bit-length output. Then, RC sends the secret key w_j to S_j via a secure channel such as presenting it face-to-face or using a public-key encryption protocol to process it before sending it.

Table 4.2.1 The notations used in Tsaur et al.'s protocol

Notation	***Definition***
$E_s(\cdot)$	the encryption function with secret key s
$D_s(\cdot)$	the decryption function with secret key s
$\oplus$	the bitwise exclusive-or operator
$\|$	the concatenation operator
$h(\cdot)$	a one-way and collision-free hash function
RC	the registration center
S_j	the jth server
U_i	the ith user
x	the secret key of the registration center
SID_j	the jth server's identity
UID_i	the ith user's identity
w_j	the secret key shared between RC and S_j
PW_i	the ith user's password
E_T_{ij}	the service period of S_j for U_i
v_i, μ_i	U_i's secret information
v_{ij}	the secret key shared between U_i and S_j
A_{ij}	the authentication parameter for U_i to log in to S_j
ru_k	a kth random value chosen by the smart card
M_{ij}	an authentication message for U_i to log in to S_j
rs_k	the kth random value chosen by S_j
sk_k	the kth session key
T	a timestamp

4.2.1 Registration Phase

Suppose user U_i wishes to access service granted from $S = \{S_1, S_2, \ldots, S_r\}$, and assume the service periods of these servers $S_1, S_2, \ldots, S_r$ for U_i are $E\_T_{i1}, E\_T_{i2}, \ldots, E\_T_{ir}$, respectively. U_i first chooses his/her identity UID_i and password PW_i, and then sends them to RC for registration via a secure channel. After receiving the message, as shown in Figure 4.2.1, RC will perform the following steps:

(1) Compute U_i's secret information $v_i = h(x + 1 \| UID_i)$ and $\mu_i = v_i \oplus h(PW_i)$.

(2) Compute the secret key $v_{ij} = h(v_i \| SID_j)$ to be shared between U_i and S_j for all $S_j \in S$.

(3) Calculate $A_{ij} = E_{w_j \oplus E\_T_{ij}}(v_{ij})$ for all $S_j \in S$.

(4) Store UID_i, μ_i, E_T_{ij} and A_{ij} to the memory of a smart card and issue this smart card to U_i.

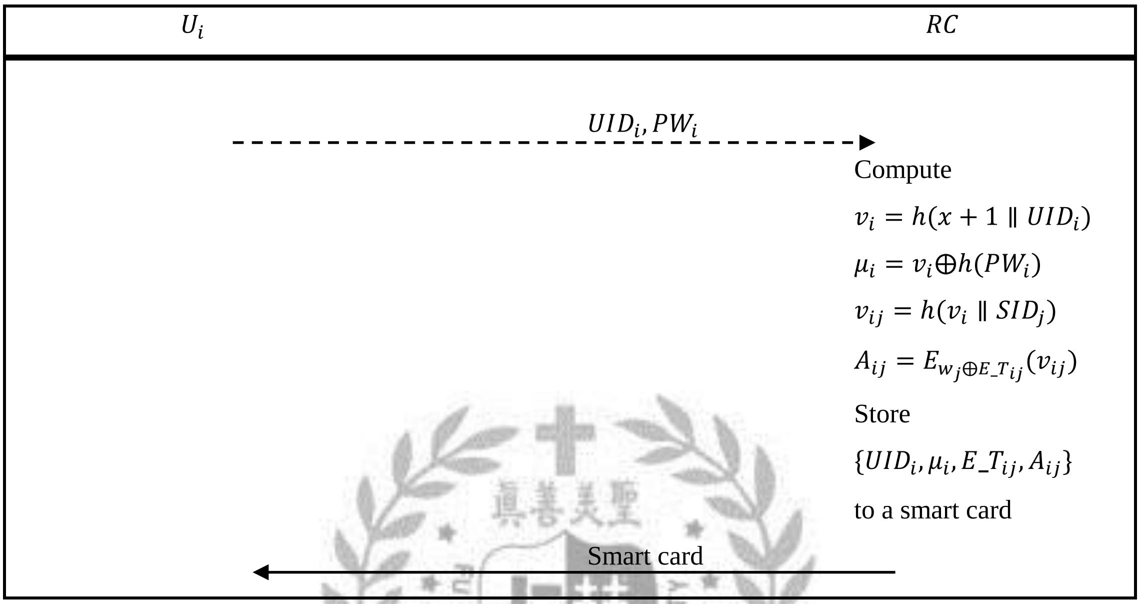

U_i
RC
UID_i, PW_i
Compute
$v_i = h(x + 1 \parallel UID_i)$
$\mu_i = v_i \oplus h(PW_i)$
$v_{ij} = h(v_i \parallel SID_j)$
$A_{ij} = E_{w_j \oplus E\_T_{ij}}(v_{ij})$
Store
$\{UID_i, \mu_i, E\_T_{ij}, A_{ij}\}$
to a smart card
Smart card

Figure 4.2.1 Registration phase of Tsaur et al.'s protocol

4.2.2 Log-in and Session Key Agreement Phase

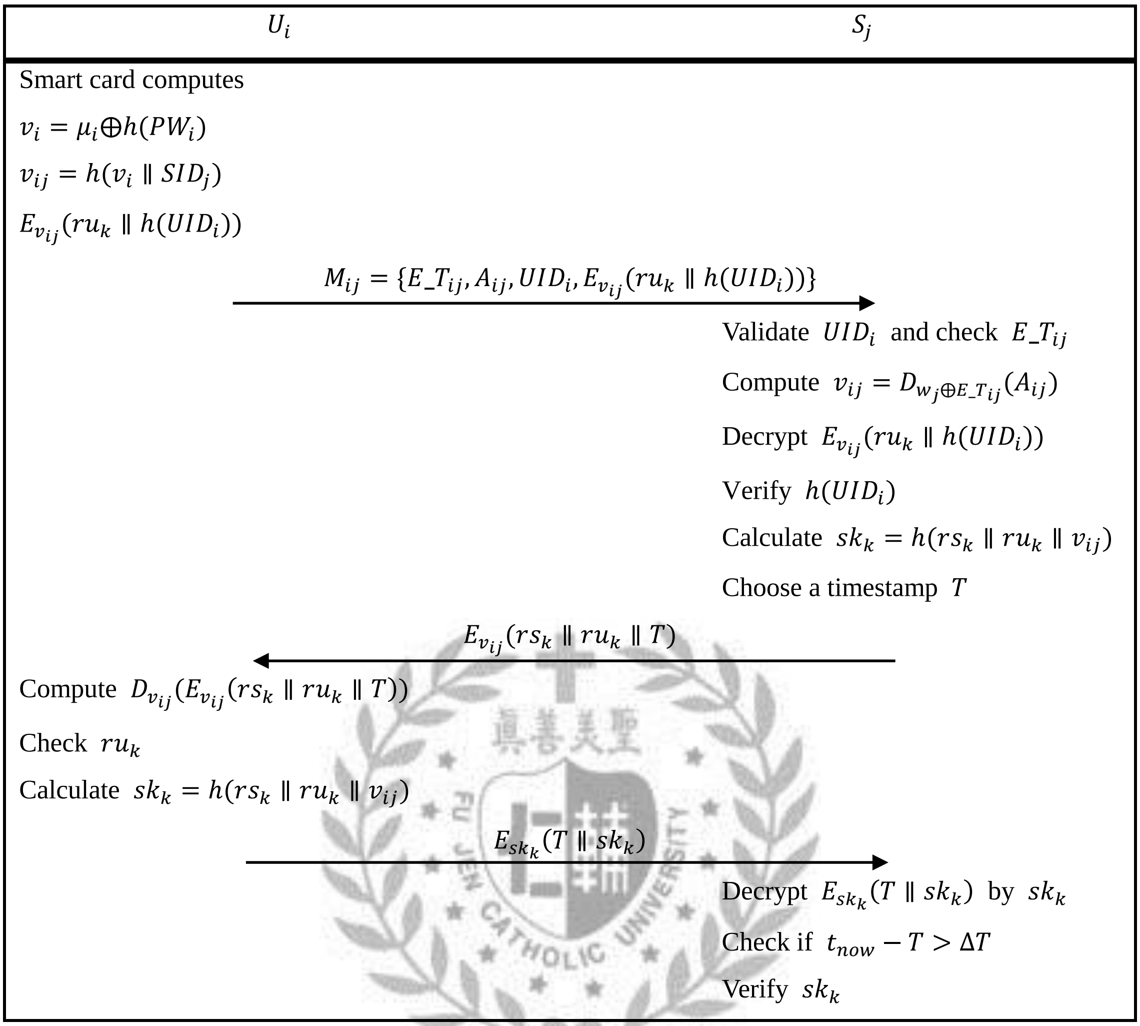
In this phase, when user U_i wishes to log in to the server S_j, U_i first inserts his/her own smart card to a card reader and keys in the password PW_i. As shown in Figure 4.2.2, the smart card and S_j will perform the following steps:

(1) The smart card first computes $v_i = \mu_i \oplus h(PW_i)$ and $v_{ij} = h(v_i \parallel SID_j)$. Then it chooses a kth random value ru_k larger than 160 bits and calculates $E_{v_{ij}}(ru_k \parallel h(UID_i))$ when U_i launches the kth log-in. A message $M_{ij} = \{E\_T_{ij}, A_{ij}, UID_i, E_{v_{ij}}(ru_k \parallel h(UID_i))\}$ is constructed and will be transmitted to S_j.

(2) After receiving M_{ij}, S_j validates the format of UID_i. If it is invalid, then S_j rejects the log-in request; otherwise, the service period E_T_{ij} is further checked to see whether it has expired. If E_T_{ij} expires, S_j will terminate the service for U_i; otherwise, S_j obtains v_{ij} by computing $D_{w_j \oplus E\_T_{ij}}(A_{ij})$. By employing v_{ij} to decrypt $E_{v_{ij}}(ru_k \parallel h(UID_i))$, S_j then obtains ru_k and $h(UID_i)$. S_j will reject this log-in if the authentication tag $h(UID_i)$ is not valid; otherwise, S_j chooses a kth random value rs_k and calculates the kth session key $sk_k = h(rs_k \parallel ru_k \parallel v_{ij})$. Then, S_j sends $E_{v_{ij}}(rs_k \parallel ru_k \parallel T)$ to U_i, where T is a timestamp chosen by S_j according to S_j's current date and time. In fact, S_j can directly adopt the time-related function of any programming language to pick up the timestamp T.

(3) Upon receiving $E_{v_{ij}}(rs_k \parallel ru_k \parallel T)$, the smart card first decrypts the message by computing $D_{v_{ij}}(E_{v_{ij}}(rs_k \parallel ru_k \parallel T))$, and then checks the correctness of ru_k. If the result is positive, the smart card computes a kth session key $sk_k = h(rs_k \parallel ru_k \parallel v_{ij})$ and the ciphertext $E_{sk_k}(T \parallel sk_k)$, and then it sends $E_{sk_k}(T \parallel sk_k)$ to S_j; otherwise, this connection will be dropped.

(4) After receiving $E_{sk_k}(T \parallel sk_k)$, S_j decrypts it with the session key sk_k, and then checks whether t_{now} is too much time behind the timestamp T by examining if $t_{now} - T > \Delta T$, where t_{now} represents S_j's current date and time, and ΔT is the biggest endurable transmission delay from S_j to U_i and then back to S_j. If not, S_j further checks the session key sk_k derived from decrypting the message $E_{sk_k}(T \parallel sk_k)$ for correctness. If the session key is correct, both U_i and S_j can use the session key sk_k for securing subsequent communication.



U_i S_j

Smart card computes

$v_i = \mu_i \oplus h(PW_i)$

$v_{ij} = h(v_i \parallel SID_j)$

$E_{v_{ij}}(ru_k \parallel h(UID_i))$

$M_{ij} = \{E\_T_{ij}, A_{ij}, UID_i, E_{v_{ij}}(ru_k \parallel h(UID_i))\}$ →

Validate UID_i and check E_T_{ij}

Compute $v_{ij} = D_{w_j \oplus E\_T_{ij}}(A_{ij})$

Decrypt $E_{v_{ij}}(ru_k \parallel h(UID_i))$

Verify $h(UID_i)$

Calculate $sk_k = h(rs_k \parallel ru_k \parallel v_{ij})$

Choose a timestamp T

← $E_{v_{ij}}(rs_k \parallel ru_k \parallel T)$

Compute $D_{v_{ij}}(E_{v_{ij}}(rs_k \parallel ru_k \parallel T))$

Check ru_k

Calculate $sk_k = h(rs_k \parallel ru_k \parallel v_{ij})$

$E_{sk_k}(T \parallel sk_k)$ →

Decrypt $E_{sk_k}(T \parallel sk_k)$ by sk_k

Check if $t_{now} - T > \Delta T$

Verify sk_k

Figure 4.2.2 Log-in and session key agreement phase of Tsaur et al.'s protocol

4.3 Cryptanalysis of Tsaur et al.'s Protocol

As we mentioned earlier, Tsaur et al.'s protocol has the following security flaws: (1) inability to resist the privileged insider attack, (2) inability to resist the known-plaintext attack, (3) inability to provide user anonymity, and (4) lack of perfect forward secrecy. Let's look into the details of these problems now.

4.3.1 Privileged Insider Attack

In the real world, it is common practice for people to simply use the same password to access all the different applications or servers so as to save the trouble of keeping a long list of passwords [16, 18]. Nevertheless, in case a malicious privileged member inside the registration center gets the password of a user U_i during the registration phase, then the malicious privileged insider may probably use that password and try to impersonate U_i in accessing various applications and servers where U_i could be a registered user. In Tsaur et al.'s protocol, the user U_i sends his/her own real identity UID_i and password PW_i directly to the registration center RC in the registration phase, which gives any possible malicious privileged insider a good chance to get U_i's password and use it to impersonate U_i around and actually succeed in accessing other applications or servers where U_i is a registered user with the same password.

4.3.2 Inability to Provide User Anonymity

In Tsaur et al.'s protocol, users send their own real identity ID_i to the server S_i directly in the log-in and session key agreement phase. This way, an adversary can easily obtain the real identity of any user by intercepting the message $M_{ij} = \{E\_T_{ij}, A_{ij}, UID_i, E_{v_{ij}}(ru_k \parallel h(UID_i))\}$ transmitted between the user and the server. In other words, Tsaur et al.'s protocol supports no user anonymity.

4.3.3 Known-plaintext Attack

A known plaintext attack is a cryptanalytic attack that takes effect when the cryptanalyst possesses a substantial quantity of corresponding plaintext and ciphertext

[11]. In Tsaur et al.'s protocol, a malicious valid user can naturally get A_{ij}, E_T_{ij}, and v_{ij} from his/her own smart card and then may use these values to derive the secret key w_j from the equation $A_{ij} = E_{w_j \oplus E\_T_{ij}}(v_{ij})$. With that, the malicious user is capable of using w_j to modify the service period E_T_{ij} and extend it. This means Tsaur et al.'s protocol is vulnerable to the known-plaintext attack.

4.3.4 Perfect Forward Secrecy

Perfect forward secrecy means the adversary can in no way derive any other session keys even if a session key or long-term key is compromised one way or another [42, 84]. In Tsaur et al.'s protocol, the smart card and the server S_j use the same secret key v_{ij} to encrypt the random values ru_k and rs_k, respectively. Unfortunately, if the secret key v_{ij} is known to an adversary, then he/she can use it to compute the kth session key $sk_k = h(rs_k \parallel ru_k \parallel v_{ij})$ for each communication session. That is to say, Tsaur et al.'s protocol cannot provide perfect forward secrecy.

4.4 Our improved Multi-server Authentication Protocol

In this section, we shall present an improved multi-server authentication protocol based on extended chaotic maps to solve the security problems that trouble Tsaur et al.'s protocol. In our protocol, the registration center RC first selects a random number X, two random integers (r, s), and a secret key $w = h(r \parallel s)$ to be shared between RC and S_j, and then RC computes $R \equiv T_w(X) \bmod p$. RC keeps the master secret keys (r, s) in secrecy and sends w to S_j via a secure channel. As with the original Tsaur protocol, there are two phases, namely the registration phase and the log-in and session key agreement phase, in our improved protocol. These two phases will be detailed right below and illustrated in Figure 4.4.1 and Figure 4.4.2. The notations used in this section are summarized in Table 4.4.1.

Table 4.4.1 The notations used in this section

Notation	***Definition***
U_i	the ith user
S_j	the jth server
RC	the registration center
PW_i	the ith user's password
ID_i	the ith user's identity
w	the secret key shared between RC and S_j
P_i	the service period of S_j for U_i
p	a large prime number that $p = 2^{130} + 7$
r, s	the secret keys of RC
X	the random number chosen by RC
R	the public key of RC, where $R \equiv T_w(X) \bmod p$
γ_i, γ_j	two random integers
$h(\cdot)$	a secure one-way hash function
$\oplus$	the exclusive-or (XOR) operation
$\|$	the concatenation operation

4.4.1 Registration Phase

When user U_i wishes to access service granted from $S = \{S_1, S_2, \dots, S_r\}$, U_i first chooses his/her identity ID_i, password PW_i, and a random number N, and then U_i sends $\{ID_i, h(PW_i) \oplus N\}$ to RC for registration via a secure channel. After receiving the message, as shown in Figure 4.4.1, RC will perform the following steps:

(1) Compute U_i 's secret information $v_i = h(ID_i \| P_i \| w)$ and $\mu_i = v_i \oplus h(PW_i) \oplus N$.

(2) Store $ID_i, \mu_i, P_i, RPUB, X, h(\cdot)$ and p to the memory of a smart card and issue this smart card to U_i.

(3) U_i computes $\mu_i' = \mu_i \oplus N$ and replaces μ_i with μ_i' in the smart card.

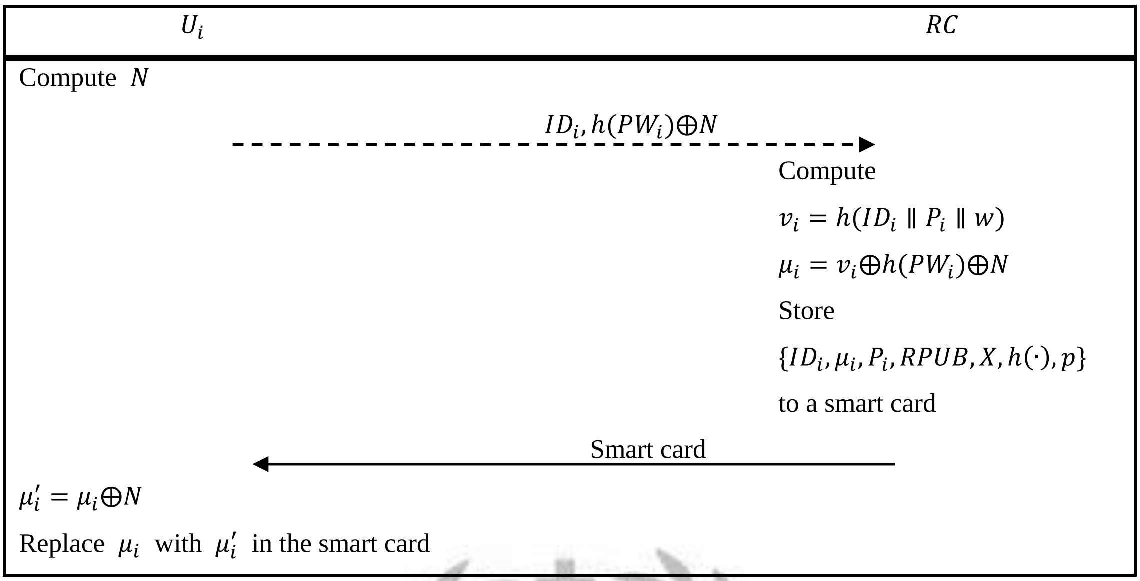

U_i
RC
Compute N
$ID_i, h(PW_i) \oplus N$
Compute
$v_i = h(ID_i \parallel P_i \parallel w)$
$\mu_i = v_i \oplus h(PW_i) \oplus N$
Store
$\{ID_i, \mu_i, P_i, RPUB, X, h(\cdot), p\}$
to a smart card
Smart card
$\mu_i' = \mu_i \oplus N$
Replace μ_i with μ_i' in the smart card

Figure 4.4.1 Registration phase of our improved multi-server authentication protocol

4.4.2 Log-in and Session Key Agreement Phase

In this phase, when user U_i wishes to log in to the server S_j, U_i first inserts his/her own smart card to a card reader and keys in the password PW_i. As shown in Figure 4.4.2, the smart card and S_j will perform the following steps:

(1) The smart card first computes $v_i = \mu_i' \oplus h(PW_i)$ and picks a random integer γ_i. Then it calculates the following:

$C_1 \equiv T_{\gamma_i}(X) \bmod p$,

$C_2 \equiv T_{\gamma_i}(R) \bmod p$,

$UID_i = ID_i \oplus h(C_1 \parallel C_2)$,

$M_{ij} = h(ID_i \parallel UID_i \parallel P_i \parallel v_i \parallel C_1 \parallel C_2)$.

A message $\{M_{ij}, UID_i, C_1, P_i\}$ is constructed and will be transmitted to S_j.

(2) After receiving $\{M_{ij}, UID_i, C_1, P_i\}$, S_j checks the equation $h(ID_i' \parallel UID_i \parallel P_i \parallel v_i' \parallel C_1 \parallel C_2')? = M_{ij}$ by computing the following:

$C_2' \equiv T_w(C_1) \bmod p$,

$ID_i' = UID_i \oplus h(C_1 \parallel C_2')$,

$v_i' = h(ID_i' \parallel P_i \parallel w)$.

If the equation above does not hold, then S_j rejects the log-in request; otherwise, the service period P_i is further checked to see whether it has expired. If P_i expires, S_j will terminate the service provided for U_i; otherwise, S_j will update the service period P_i with $P_i^{new} = P_i - 1$, and then compute the new secret information $v_i^{new} = h(ID_i' \parallel P_i^{new} \parallel w)$ for U_i. S_j then calculates $V_i = v_i' \oplus v_i^{new}$ to protect v_i^{new} and chooses a random integer γ_j. By using the random integer γ_j, S_j computes $C_3 \equiv T_{\gamma_j}(X) \bmod p$ and the session key $SK \equiv T_{\gamma_j}(C_1) \equiv T_{\gamma_j \gamma_i}(X) \bmod p$. Finally, S_j computes $M_{ji} = h(ID_i' \parallel v_i' \parallel v_i^{new} \parallel P_i^{new} \parallel C_2' \parallel C_3 \parallel SK)$ and sends the message $\{M_{ji}, C_3, V_i\}$ to U_i.

(3) Upon receiving the message $\{M_{ji}, C_3, V_i\}$ from S_j, the smart card checks the equation $h(ID_i \parallel v_i \parallel v_i^{new\prime} \parallel P_i^{new} \parallel C_2 \parallel C_3 \parallel SK')? = M_{ji}$ by doing the following calculation:

$v_i^{new\prime} = V_i \oplus v_i$,

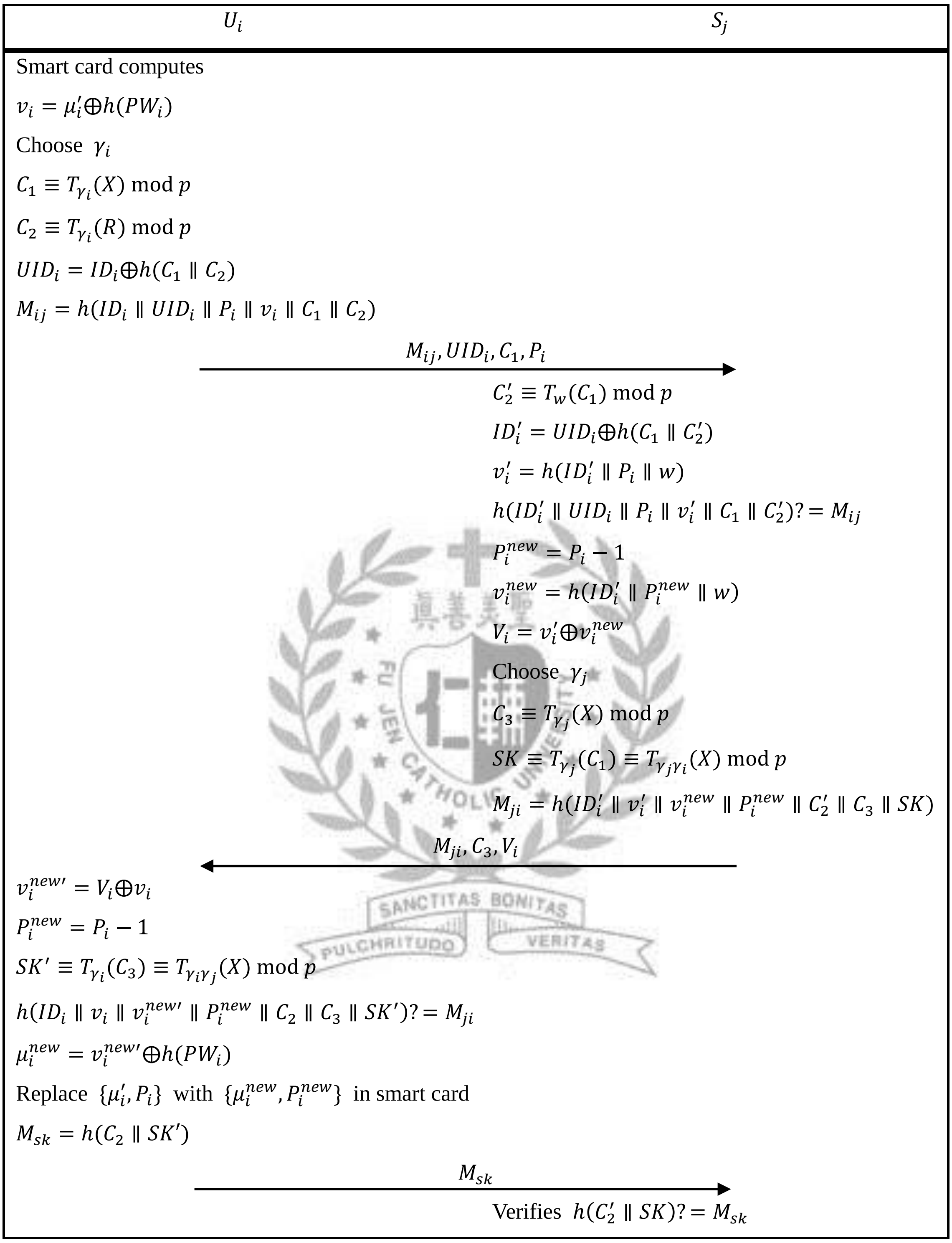
$P_i^{new} = P_i - 1$,

$SK' \equiv T_{\gamma_i}(C_3) \equiv T_{\gamma_i \gamma_j}(X) \bmod p$.

If the equation above holds, the smart card computes $\mu_i^{new} = v_i^{new\prime} \oplus h(PW_i)$ and replaces $\{\mu_i', P_i\}$ with $\{\mu_i^{new}, P_i^{new}\}$; otherwise, this connection will be dropped.

Finally, the smart card computes $M_{sk} = h(C_2 \parallel SK')$ and transmits it to the server S_j.

(4) Upon receiving the message $\{M_{sk}\}$ from U_i, S_j checks the correctness of the session key SK by confirming if the equation $h(C_2' \parallel SK)? = M_{sk}$ holds. If the session key is correct, both U_i and S_j can use SK for securing a subsequent session of communication. Otherwise, this connection will be dropped.



U_i S_j

Smart card computes

$v_i = \mu_i' \oplus h(PW_i)$

Choose γ_i

$C_1 \equiv T_{\gamma_i}(X) \bmod p$

$C_2 \equiv T_{\gamma_i}(R) \bmod p$

$UID_i = ID_i \oplus h(C_1 \parallel C_2)$

$M_{ij} = h(ID_i \parallel UID_i \parallel P_i \parallel v_i \parallel C_1 \parallel C_2)$

$$\xrightarrow{M_{ij}, UID_i, C_1, P_i}$$

$C_2' \equiv T_w(C_1) \bmod p$

$ID_i' = UID_i \oplus h(C_1 \parallel C_2')$

$v_i' = h(ID_i' \parallel P_i \parallel w)$

$h(ID_i' \parallel UID_i \parallel P_i \parallel v_i' \parallel C_1 \parallel C_2')? = M_{ij}$

$P_i^{new} = P_i - 1$

$v_i^{new} = h(ID_i' \parallel P_i^{new} \parallel w)$

$V_i = v_i' \oplus v_i^{new}$

Choose γ_j

$C_3 \equiv T_{\gamma_j}(X) \bmod p$

$SK \equiv T_{\gamma_j}(C_1) \equiv T_{\gamma_j \gamma_i}(X) \bmod p$

$M_{ji} = h(ID_i' \parallel v_i' \parallel v_i^{new} \parallel P_i^{new} \parallel C_2' \parallel C_3 \parallel SK)$

$$\xleftarrow{M_{ji}, C_3, V_i}$$

$v_i^{new'} = V_i \oplus v_i$

$P_i^{new} = P_i - 1$

$SK' \equiv T_{\gamma_i}(C_3) \equiv T_{\gamma_i \gamma_j}(X) \bmod p$

$h(ID_i \parallel v_i \parallel v_i^{new'} \parallel P_i^{new} \parallel C_2 \parallel C_3 \parallel SK')? = M_{ji}$

$\mu_i^{new} = v_i^{new'} \oplus h(PW_i)$

Replace $\{\mu_i', P_i\}$ with $\{\mu_i^{new}, P_i^{new}\}$ in smart card

$M_{sk} = h(C_2 \parallel SK')$

$$\xrightarrow{M_{sk}}$$

Verifies $h(C_2' \parallel SK)? = M_{sk}$

Figure 4.4.2 Log-in and session key agreement phase of our improved multi-server authentication protocol

4.5 Authentication Proof Based on BAN Logic

The BAN-logic is one of the most celebrated methods in the field of cryptographic protocol analysis. It makes it possible to formally analyze cryptographic protocols in a simple way [4, 76]. To use the BAN logic, we first define the basic notations, goals, and assumptions. The details are shown as follows.

4.5.1 Notations

First of all, let’s notice the syntax of the BAN logic. We define A, B as participators and X as a formula, and use some instances to exam the syntax and notations of the BAN logic [4, 76].

- $A|\equiv X$: A believes X is true.
- $A \triangleleft X$: A sees or holds X.
- $A|\equiv B$: A believes B’s actions, e.g., $A|\equiv B \triangleleft X$ means that A believes B holds X.
- $A|\Rightarrow X$: A has complete control over X. This can be used to denote a certificate authority.
- $A|\sim X$: A once said X.
- $\#(X)$: X is fresh, which means X is recent or X is a nonce.
- $A \overset{X}{\leftrightarrow} B$: X is a secret key or secret information shared between A and B.
- $\overset{X}{\mapsto} A$ and X^{-1}: A has a public key X and a private (secret) key X^{-1}.
- $\{M\}_X$: Plain text M is encrypted by X.

- (X, Y): X or Y is one part of formula (X, Y).

- $\frac{Rule\ 1}{Rule\ 2}$: We can infer $Rule\ 2$ from $Rule\ 1$, e.g., $\frac{A \text{ creates random } X}{A \mid\equiv \#(X)}$ means that A creates X, so A believes X is fresh.

We use the BAN logic to transform our protocol, as illustrated in Figure 4.4.2, into an idealized form. The messages in the idealized form are as follows:

M1. $U_i \longrightarrow S_j: h\big(UID_i, P_i, v_i, \{X\}_{\gamma_i}, \{X\}_{w\cdot\gamma_i}\big), UID_i, \{X\}_{\gamma_i}, P_i$

M2. $S_j \longrightarrow U_i: h\left(v_i', v_i^{new}, P_i^{new}, \{X\}_{w\cdot\gamma_i}, \{X\}_{\gamma_j}, U_i \overset{SK}{\leftrightarrow} S_j\right), \{X\}_{\gamma_j}, V_i$

M3. $U_i \longrightarrow S_j: h(\{X\}_{\gamma_i}, U_i \overset{SK}{\leftrightarrow} S_j)$

4.5.2 Goals

The goals of our proposed protocol are to be stated in the syntax of the Ban logic here. Legal user U_i, legal user U_j, and the trusted authority TA are the participators in the proposed protocol. The phase-I access control of our protocol has the following two goals: that U_i believes U_j is a legal user, and that U_j believes U_i is a legal user. The goals of our protocol are shown as formula G1 and G2 in the language of the BAN logic.

G1. $U_i \mid\equiv U_i \overset{SK}{\leftrightarrow} S_j$

G2. $S_j \mid\equiv U_i \overset{SK}{\leftrightarrow} S_j$

G3. $U_i \mid\equiv S_j \mid\equiv U_i \overset{SK}{\leftrightarrow} S_j$

G4. $S_j \mid\equiv U_i \mid\equiv U_i \overset{SK}{\leftrightarrow} S_j$

4.5.3 Assumptions

In order to analyze our protocol by using the BAN logic, we have made some assumptions as follows:

A1. $U_i|\equiv\#(\gamma_i)$

A2. $S_j|\equiv\#(\gamma_j)$

A3. $U_i|\equiv S_j \overset{w}{\leftrightarrow} RC$

A4. $S_j|\equiv S_j \overset{w}{\leftrightarrow} RC$

A5. $U_i|\equiv S_j|\equiv S_j \overset{w}{\leftrightarrow} RC$

A6. $S_j|\equiv U_i|\equiv S_j \overset{w}{\leftrightarrow} RC$

A7. $U_i|\equiv S_j|\Longrightarrow U_i \overset{SK}{\leftrightarrow} S_j$

A8. $S_j|\equiv U_i \overset{SK}{\leftrightarrow} S_j$

4.5.4 Verification

This subsection shows the correctness of our protocol confirmed by analyzing the idealized form of our protocol using the assumptions above and the rules of the BAN logic. The main steps of the proof are as follows:

U_i chooses random γ_i

V1. $U_i|\equiv\gamma_i$

V2. $U_i|\equiv\#(\gamma_i)$

Message 1: $U_i \longrightarrow S_j: h\big(\{X\}_{\gamma_i}, \{X\}_{w\cdot\gamma_i}\big), \{X\}_{\gamma_i}$

V3. $S_j \lhd h\big(\{X\}_{\gamma_i}, \{X\}_{w\cdot\gamma_i}\big), \{X\}_{\gamma_i}$

V4. $\frac{S_j \lhd h\big(\{X\}_{\gamma_i}, \{X\}_{w\cdot\gamma_i}\big), \{X\}_{\gamma_i}}{S_j|\equiv U_i|\sim\{X\}_{\gamma_i}}$

S_j chooses random γ_j

V5. $S_j|\equiv\gamma_j$

V6. $S_j|\equiv\#(\gamma_j)$

S_j computes the session key $U_i \overset{SK}{\leftrightarrow} S_j = \{X\}_{\gamma_i \cdot \gamma_j}$

Message 2: $S_j \longrightarrow U_i : h\left(\{X\}_{w\cdot\gamma_i}, \{X\}_{\gamma_j}, U_i \overset{SK}{\leftrightarrow} S_j\right), \{X\}_{\gamma_j}$

V7. $U_i \lhd h\left(\{X\}_{w\cdot\gamma_i}, \{X\}_{\gamma_j}, U_i \overset{SK}{\leftrightarrow} S_j\right), \{X\}_{\gamma_j}$

V8. $\frac{U_i \lhd \gamma_i, U_i \lhd \{X\}_{\gamma_j}}{U_i \lhd U_i \overset{SK}{\longleftrightarrow} S_j}$

V9. $\frac{U_i \lhd h\left(\{X\}_{w\cdot\gamma_i}, \{X\}_{\gamma_j}, U_i \overset{SK}{\longleftrightarrow} S_j\right), \{X\}_{\gamma_j}, S_j|\equiv U_i|\sim\{X\}_{\gamma_i}}{U_i|\equiv S_j|\sim(\{X\}_{w\cdot\gamma_i}, \{X\}_{\gamma_j}, U_i \overset{SK}{\longleftrightarrow} S_j)}$

V10. $\frac{U_i|\equiv\#(\gamma_i), U_i|\equiv S_j|\sim(\{X\}_{w\cdot\gamma_i}, \{X\}_{\gamma_j}, U_i \overset{SK}{\longleftrightarrow} S_j)}{U_i|\equiv S_j|\equiv(\{X\}_{\gamma_i}, \{X\}_{\gamma_j}, U_i \overset{SK}{\longleftrightarrow} S_j)}$

V11. $\frac{U_i|\equiv S_j|\equiv(\{X\}_{\gamma_i}, \{X\}_{\gamma_j}, U_i \overset{SK}{\longleftrightarrow} S_j)}{U_i|\equiv S_j|\equiv U_i \overset{SK}{\longleftrightarrow} S_j}$

V12. $\frac{U_i|\equiv S_j|\Longrightarrow U_i \overset{SK}{\longleftrightarrow} S_j, U_i|\equiv S_j|\equiv(\{X\}_{\gamma_i}, \{X\}_{\gamma_j}, U_i \overset{SK}{\longleftrightarrow} S_j)}{U_i|\equiv U_i \overset{SK}{\longleftrightarrow} S_j}$

Message 3: $U_i \longrightarrow S_j : h(\{X\}_{\gamma_i}, U_i \overset{SK}{\leftrightarrow} S_j)$

V13. $S_j \lhd h(\{X\}_{\gamma_i}, U_i \overset{SK}{\leftrightarrow} S_j)$

V14. $\frac{S_j|\equiv\#(\gamma_j)}{S_j|\equiv\#(U_i \overset{SK}{\longleftrightarrow} S_j)}$

V15. $\frac{S_j \lhd h\left(\{X\}_{\gamma_i}, U_i \overset{SK}{\longleftrightarrow} S_j\right), S_j|\equiv U_i \overset{SK}{\longleftrightarrow} S_j}{S_j|\equiv U_i|\sim U_i \overset{SK}{\longleftrightarrow} S_j}$

V16. $\frac{S_j|\equiv\#(U_i \overset{SK}{\longleftrightarrow} S_j), S_j|\equiv U_i|\sim U_i \overset{SK}{\longleftrightarrow} S_j}{S_j|\equiv U_i|\equiv U_i \overset{SK}{\longleftrightarrow} S_j}$

As a result, inferring from formula A8, V11, V12 and V16, we can now be sure that our new protocol is truly capable of achieving the goals.

4.6 Security Analysis of our improved Protocol

In order to prove that our improved protocol does not have any of the flaws Tsaur et al.'s protocol has that we pointed out earlier, we will make sure that our protocol does support user anonymity, mutual authentication, and perfect forward secrecy. Besides that, we shall also test our protocol against various possible attacks including the privileged insider attack, replay attack, known-plaintext attack, and Bergamo et al.'s attack.

4.6.1 Privileged Insider Attack

In the registration phase of our improved protocol, the user U_i sends $\{ID_i, h(PW_i) \oplus N\}$ to the registration center RC. A malicious privileged insider has no way to derive U_i's password and use it to impersonate U_i because he-she cannot obtain the random number N. Therefore, the privileged insider attack poses no threat to our improved protocol.

4.6.2 User Anonymity

Suppose an adversary has eavesdropped the communication between a user U_i and the server S_j, he/she may try to trace U_i's real identity and gather confidential information about U_i. In our improved protocol, the real identity of U_i is protected by the encrypted message $UID_i = ID_i \oplus h(C_1 \parallel C_2)$. If the adversary wanted to derive ID_i

from UID_i, he/she would have to face the DLP problem. In other words, we can say that our improved protocol does provide the user with high level anonymity.

4.6.3 Mutual Authentication

In the log-in and session key agreement phase of our protocol, upon receiving the message $\{M_{ij}, UID_i, C_1, P_i\}$ from a user U_i, the server S_j checks the validity of $h(ID_i' \parallel UID_i \parallel P_i \parallel v_i' \parallel C_1 \parallel C_2')? = M_{ij}$. If the equation holds, S_j considers U_i a legal user. Then S_j computes $M_{ji} = h(ID_i' \parallel v_i' \parallel v_i^{new} \parallel P_i^{new} \parallel C_2' \parallel C_3 \parallel SK)$ and sends the message $\{M_{ji}, C_3, V_i\}$ to U_i. Likewise, upon receiving the message $\{M_{ji}, C_3, V_i\}$ from S_j, U_i checks the validity of $h(ID_i \parallel v_i \parallel v_i^{new\prime} \parallel P_i^{new} \parallel C_2 \parallel C_3 \parallel SK')? = M_{ji}$. If it holds, U_i considers S_j a legal server. Since only the registration center RC and the server S_j know the secret key w, U_i and S_j store the value $v_i = h(ID_i \parallel P_i \parallel w)$ and w, respectively. Finally, both U_i and S_j generate the same session key SK. This means our improved protocol does offer perfect mutual authentication between U_i and S_j, and therefore it is secure against the impersonation attack.

4.6.4 Replay Attack

A replay attack is a form of network attack where a valid chunk of data transmission is maliciously or fraudulently repeated or delayed. The replay attack will fail in the attempt to break our improved protocol because the freshness of the messages transmitted is provided by the random nonces γ_i and γ_j. Except for U_i (or S_j), only S_j (or U_i)

can embed the shared common session key SK and the secret value C_2 in the message M_{ij} (or M_{ji}).

4.6.5 Known-plaintext Attack

In Tsaur et al.'s protocol, a malicious valid user can extract A_{ij}, E_T_{ij}, and v_{ij} from his/her own smart card, and then these values can be used to derive the secret key w_j by applying the equation $A_{ij} = E_{w_j \oplus E\_T_{ij}}(v_{ij})$. After that, the malicious user can use w_j to modify the service period E_T_{ij} and extend it. In our improved protocol, an adversary may be able to obtain X, C_1 and C_3 easily, but there is no way to derive γ_i and γ_j from those values. The reason is that everything has been encrypted by applying Chebyshev polynomials and are only known to the user and the server. Moreover, what we use in our improved protocol is not the ordinary Chebyshev polynomials but the enhanced Chebyshev polynomials, where the periodicity of the cosine function is avoided by extending the interval of Y to $(-\infty, +\infty)$, not to mention that the service period P_i of any valid user is encrypted by applying $v_i = h(ID_i \parallel P_i \parallel w)$. Therefore, we conclude that the known-plaintext attack can do no damage to our improved protocol.

4.6.6 Perfect Forward Secrecy

Perfect forward secrecy means even if a session key or long-term key is compromised one way or another, the adversary still has no way to derive any of the other session keys from the cracked one [16, 17]. In our improved protocol, the smart card and server S_j use the random numbers γ_i and γ_j to compute the current session key $SK \equiv T_{\gamma_j \gamma_i}(X) \bmod p$. Should the current session key SK be somehow known to an adversary, he/she would still be unable to use it to compute any of the other session keys $SK \equiv T_{\delta_j\delta_i}(X) \bmod p$ since the random numbers are different in each communication session. This is the way our improved protocol ensures perfect forward secrecy.

4.6.7 Bergamo et al.'s Attack

Bergamo et al.'s attack [3] works on the condition that an adversary can obtain the related elements X, C_1, and C_3 and derive γ_i and γ_j from them. In our improved protocol, the adversary may be able to obtain X, C_1 and C_3 easily, but there is no way to derive γ_i and γ_j from those values. The reason is that the elements are encrypted by Chebyshev polynomials and are only known to the user and the server. Moreover, our protocol utilizes the enhanced Chebyshev polynomials, where the periodicity of the cosine function is avoided by extending the interval of Y to $(-\infty, +\infty)$. Hence, Bergamo et al.'s attack will take no effect in cracking our improved protocol.

4.7 Performance Discussion and Comparison

In this section, we discuss the performance of our improved protocol. The security properties of our improved protocol, Juang's protocol [25], Tsai's protocol [68], Li et al.'s protocol [47], and Tsaur et al.'s protocol [70] have been compared, and the results are shown in Table 4.7.1. Obviously, we can see that our improved protocol has a higher level of security than the other protocols. Besides, we have also compared the computational primitives involved in both the registration phase and the log-in and session key agreement phase of our improved protocol with those of the same other protocols. The results are presented in Table 4.7.2.

As Table 4.7.1 and Table 4.7.2 show, even though the computation costs of Li et al.'s protocol and Tsaur et al.'s protocol are slightly lower than that of our improved protocol, their light weights come along with security flaws, which is a sacrifice we believe no system constructors in their senses would make. By contrast, due to the use of hash functions and Chebyshev chaotic maps, our improved protocol is capable of offering thorough security protection at a very reasonable computation cost, exhibiting performance of high efficiency.

Table 4.7.1 Security comparisons among ours and other related protocols

Comparative protocols → Security requirements ↓	Juang (2004)	Tsai (2008)	Li et al. (2013)	Tsaur et al. (2012)	Our protocol
Privileged insider attack	✗	✗	✗	✗	✓
User anonymity	✗	✗	✓	✗	✓
Mutual authentication	✓	✓	✓	✓	✓
Impersonation Attack	✓	✗	✓	✗	✓
Replay attack	✓	✗	✓	✓	✓
Known-plaintext attack	✗	✓	✗	✗	✓
Perfect forward secrecy	✗	✓	✗	✗	✓
Validity proof	✗	✗	✗	✓	✓

Table 4.7.2 Performance comparisons among ours and other related protocols

Comparative protocols → Executive computations ↓	Juang (2004)	Tsai (2008)	Li et al. (2013)	Tsaur et al. (2012)	Our protocol
User registration	$1T_H$	$2T_H$	$3T_H + 1T_S$	$3T_H + 1T_S$	$2T_H$
Sever registration	$1T_H$	$1T_H$	$1T_H$	$1T_H$	$1T_H$
User authentication	$3T_H + 3T_S$	$5T_H$	$4T_H + 3T_S$	$4T_H + 3T_S$	$5T_H + 3T_C$
Sever authentication	$3T_H + 4T_S$	$3T_H$	$2T_H + 4T_S$	$2T_H + 4T_S$	$6T_H + 3T_C$
RC authentication	$1T_H + 2T_S$	$1T_H$	-	-	-
Total computations	$9T_H + 9T_S$	$12T_H$	$10T_H + 8T_S$	$10T_H + 8T_S$	$14T_H + 6T_C$

T_H: Time for performing a one-way hash function.

T_S: Time for performing a symmetric operation.

T_C: Time for performing a Chebyshev chaotic map operation.

Chapter 5 Conclusions

In this study, we proposed three user authentication and key agreement protocols based on extended chaotic maps. In Chapter 2, we presented a cryptanalysis of Das’s protocol and pointed out it security weaknesses. We have shown that Das’s protocol is vulnerable to the privileged insider attack, the off-line password guessing attack, and also cannot provide user anonymity. To remedy these weaknesses, we proposed a secure biometric-based remote user authentication with key agreement protocol using extended chaotic maps. The proposed protocol not only can resist the above-mentioned attacks, but also provide user anonymity.

In Chapter 3, we proposed a three-party password-based authenticated key exchange protocol with user anonymity using extended chaotic maps, which is more efficient and secure than previously proposed protocols. In order to enhance the efficiency and security, we used the extended chaotic maps both to encrypt and decrypt the information transmitted by either the user or server. In security and performance analysis, we have shown that our protocol is more efficient and secure than others since our protocol only uses hash operation and XOR operation. Moreover, our protocol can also provide user anonymity to guarantee the identity of users, which is transmitted in the insecure public network.

In Chapter 4, we have briefly reviewed and analyzed Tsaur et al.’s multi-server authentication protocol with key agreement. We pointed out that Tsaur et al.’s protocol is vulnerable to the privileged insider attack and the known-plaintext attack, and is unable to provide user anonymity and perfect forward secrecy. In order to remedy all the problems named, we have presented an improved multi-server authentication protocol with key agreement based on extended chaotic maps and analyzed its security and performance. Compared with several other related protocols including Tsaur et al.’s, our improved protocol obviously comes with much better security features such as mutual authentication, user anonymity, and achieves perfect forward secrecy. As for the computation cost, owing to the employment of hash operations and Chebyshev chaotic maps, the thorough security protection that our improved protocol offers only causes a slight, very reasonable increase of computation. Consequently, we can conclude that our improved protocol is both highly secure and extremely efficient.

References

[1] M. Abdalla and D. Pointcheval, "Interactive Diffie-Hellman Assumptions with Applications to Password-based Authentication," *Lecture Notes in Computer Science*, vol. 3570, pp. 341-356, 2005.

[2] S. M. Bellovin and M. Merritt, "Encrypted Key Exchange: Password-based Protocols Secure Against Dictionary Attacks," *Proceedings of IEEE Computer Society Symposium on Security and Privacy*, pp. 72-84, 1992.

[3] P. Bergamo, P. D’Arco, A. De Santis, and L. Kocarev, "Security of Public-key Cryptosystems Based on Chebyshev Polynomials," *IEEE Transactions on Circuits and Systems-I,* vol. 52, no. 7, pp. 1382-1393, 2005.

[4] M. Burrows, M. Abadi, and R. M. Needham, "A Logic of Authentication," *Proceedings of the Royal Society of London A*, vol. 426, no. 1871, pp. 233-271, 1989.

[5] C. C. Chang and S. J. Hwang, "Using Smart Cards to Authenticate Remote Passwords," *Computers & Mathematics with Applications*, vol. 26, no. 7, pp. 19-27, 1993.

[6] T. Y. Chang, M. S. Hwang, and W. P. Yang, "A Communication-efficient Three-party Password Authenticated Key Exchange Protocol," *Information Sciences*, vol. 181, no. 1, pp. 217-226, 2011.

[7] T. Y. Chang, W. P. Yang, and M. S. Hwang, "Simple Authenticated Key Agreement and Protected Password Change Protocol," *Computers & Mathematics with Applications*, vol. 49, no. 5-6, pp. 703-714, 2005.

[8] F. Dachselt and W. Schwarz, "Chaos and Cryptography," *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 48, no. 12, pp. 1498-1509, 2001.

[9] A. K. Das, "Analysis and Improvement on an Efficient Biometric-based Remote User Authentication Protocol Using Smart Cards," *IET Information Security*, vol. 5, no. 3, pp. 145-151, 2011.

[10] M. Deng, J. Ma, and F. Le, "Universally Composable Three Party Password-based Key Exchange Protocol," *China Communications*, vol. 6, no. 3, pp. 150-154, 2009.

[11] W. Diffie and M. Hellman, "New Directions in Cryptography," *IEEE Transactions on Information Theory*, vol. 22, no. 6, pp. 644-654, 1976.

[12] L. Fan, J. H. Li, and H. W. Zhu, "An Enhancement of Timestamp-based Password Authentication Protocol," *Computers & Security*, vol. 21, no. 7, pp. 665-667, 2002.

[13] J. Fridrich, "Symmetric Ciphers Based on Two-dimensional Chaotic Maps," *International Journal of Bifurcation and Chaos*, vol. 8, no. 6, pp. 1259-1284, 1998.

[14] S. Han and E. Chang, "Chaotic Map Based Key Agreement with/out Clock Synchronization," *Chaos, Solitons & Fractals*, vol. 39, no. 3, pp. 1283-1289, 2009.

[15] M. I. Hassan and A. Abdullah, "A New Grid Resource Discovery Framework," *The International Arab Journal of Information Technology*, vol. 8, no. 1, pp. 99-107, 2011.

[16] D. He, Y. Chen, and J. Chen, "Cryptanalysis and Improvement of an Extended Chaotic Maps-based Key Agreement Protocol," *Nonlinear Dynamics*, vol. 69, no. 3, pp. 1149-1157, 2012.

[17] D. He, M. Ma, Y. Zhang, C. Chen, and J. Bu, "A Strong User Authentication Protocol with Smart Cards for Wireless Communications," *Computer Communications*, vol. 34, no. 3, pp. 367-374, 2011.

[18] H. He, S. Wu, and J. Chen, "Note on Design of Improved Password Authentication and Update Protocol Based on Elliptic Curve Cryptography," *Mathematical and Computer Modelling*, vol. 55, no. 3-4, pp. 1661-1664, 2012.

[19] H. F. Huang, "A Simple Three-party Password-based Key Exchange Protocol," *International Journal of Communication Systems*, vol. 22, no. 7, pp. 857-862, 2009.

[20] M. S. Hwang and L. H. Li, "A New Remote User Authentication Protocol Using Smart Cards," *IEEE Transactions on Consumer Electronics*, vol. 46, no. 1, pp. 28-30, 2000.

[21] M. S. Hwang and C. Y. Liu, "Authenticated Encryption Protocols: Current Status and Key Issues," *International Journal of Network Security*, vol. 1, no. 2, pp. 61-73, 2005.

[22] C. L. Hwang and C. Y. Shih, "A Distributed Active-vision Network-space Approach for the Navigation of a Car-like Wheeled Robot," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 3, pp. 846-855, 2009.

[23] A. K. Jain, A. Ross, and S. Prabhakar, "An Introduction to Biometric Recognition," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 4-20, 2004.

[24] W. S. Juang, S. T. Chen, and H. T. Liaw, "Robust and Efficient Password-authenticated Key Agreement Using Smart Cards," *IEEE Transactions on Industrial Electronics*, vol. 55, no. 6, pp. 2551-2556, 2008.

[25] W. S. Juang, "Efficient Multi-server Password Authenticated Key Agreement Using Smart Cards," *IEEE Transactions on Consumer Electronics*, vol. 50, no. 1, pp. 251-255, 2004.

[26] M. K. Khan, J. Zhang, and X. Wang, "Chaotic Hash-based Fingerprint Biometric Remote User Authentication Protocol on Mobile Devices," *Chaos, Solitons & Fractals*, vol. 35, no. 3, pp. 519-524, 2008.

[27] L. Kocarev, "Chaos-based Cryptography: a Brief Overview," *IEEE Circuits and*

Systems Magazine, vol. 1, no. 3, pp. 6-21, 2001.

[28] L. Kocarev and Z. Tasev, "Public-key Encryption Based on Chebyshev Maps," *In Proceedings of the International Symposium on Circuits and Systems (ISCAS '03)*, vol. 3, pp. III-28-III-31, 2003.

[29] P. Kocher, J. Jaffe, and B. Jun, "Differential Power Analysis," *Lecture Notes in Computer Science*, vol. 1666, pp. 388-397, 1999.

[30] J. T. Kohl, B. C. Neuman, and Y. Theodore, "The Evolution of the Kerberos Authentication Service," *in Distributed Open Systems*, *IEEE Computer Society Press*, pp. 78-94, 1994.

[31] L. Lamport, "Password Authentication with Insecure Communication," *Communications of the ACM*, vol. 24, no. 11, pp. 770-772, 1981.

[32] C. C. Lee, R. X. Chang, and H. J. Ko, "Improving Two Novel Three-party Encrypted Key Exchange Protocols with Perfect Forward Secrecy", *International Journal of Foundations of Computer Science*, vol. 21, no. 6, pp. 979-991, 2010.

[33] C. C. Lee and Y. F. Chang, "On Security of a Practical Three-party Key Exchange Protocol with Round Efficiency", *Information Technology and Control*, vol. 37, no. 4, pp. 333-335, 2008.

[34] C. C. Lee, S. D. Chen, and C. L. Chen, "A Computation-efficient Three-party Encrypted Key Exchange Protocol," *Applied Mathematics & Information Sciences*, vol. 6, no. 3, pp. 573-579, 2012.

[35] C. C. Lee, C. L. Chen, C. Y. Wu, and S. Y. Huang, "An Extended Chaotic Maps-based Key Agreement Protocol with User Anonymity," *Nonlinear Dynamics*, vol. 69, no. 1-2, pp. 79-87, 2012.

[36] N. Y. Lee and Y. C. Chiu, "Improved Remote Authentication Protocol with Smart

Card," *Computer Standards & Interfaces*, vol. 27, no. 2, pp. 177-180, 2005.

[37] C. C. Lee and C. W. Hsu, "A Secure Biometric-based Remote User Authentication with Key Agreement Protocol Using Extended Chaotic Maps," *Nonlinear Dynamics*, vol. 71, no. 1-2, pp. 201-211, 2013.

[38] C. C. Lee, K. Y. Huang, and S. Y. Huang, "On-line Password Guessing Attack on Lu-Cao Key Agreement Protocol for Secure Authentication," *Journal of Discrete Mathematical Sciences & Cryptography*, vol. 12, no. 5, pp. 595-598, 2009.

[39] C. C. Lee, M. S. Hwang, and I. E. Liao, "Security Enhancement on a New Authentication Protocol with Anonymity for Wireless Environments," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 5, pp. 1683-1687, 2006.

[40] T. F. Lee, T. Hwang, and C. L. Lin, "Enhanced Three-party Encrypted Key Exchange Without Server Public Keys," *Computers & Security*, vol. 23, no. 7, pp. 571-577, 2004.

[41] S. W. Lee, H. S. Kim, and K. Y. Yoo, "Efficient Verifier-based Key Agreement Protocol for Three Parties Without Server's Public Key," *Applied Mathematics and Computation*, vol. 167, no. 2, pp. 996-1003, 2005.

[42] C. C. Lee, C. T. Li, and R. X. Chang, "A Simple and Efficient Authentication Protocol for Mobile Satellite Communication Systems," *International Journal of Satellite Communications and Networking*, vol. 30, no. 1, pp. 29-38, 2012.

[43] C. C. Lee, T. H. Lin, and R. X. Chang, "A Secure Dynamic ID Based Remote User Authentication Protocol for Multi-server Environment Using Smart Cards," *Expert Systems with Applications*, vol. 38, no. 11, pp. 13863-13870, 2011.

[44] C. T. Li and M. S. Hwang, "An Efficient Biometric-based Remote Authentication Protocol Using Smart Cards," *Journal of Network and Computer Applications*, vol.

33, no. 1, pp. 1-5, 2010.

[45] C. T. Li and M. S. Hwang, "An Online Biometrics-based Secret Sharing Protocol for Multiparty Cryptosystem Using Smart Cards," *International Journal of Innovative Computing, Information and Control*, vol. 6, no. 5, pp. 2181-2188, 2010.

[46] C. T. Li and C. C. Lee, "A Robust Remote User Authentication Protocol Using Smart Card," *Information Technology and Control*, vol. 40, no. 3, pp. 236-245, 2011.

[47] C. T. Li , C. C. Lee , C. Y. Weng , and C. I. Fan, "An Extended Multi-server-based User Authentication and Key Agreement Protocol with User Anonymity," *KSII Transactions on Internet and Information Systems*, vol. 7, no.1, pp. 119-131, 2013.

[48] L. H. Li, I. C. Lin, and M. S. Hwang, "A Remote Password Authentication Protocol for Multi-server Architecture Using Neural Networks," *IEEE Transactions on Neural Networks*, vol. 12, no. 6, pp. 1498-1504, 2001.

[49] J. P. Lin and J. M. Fu, "Authenticated Key Agreement Protocol with Privacy-protection in the Three-party Setting," *International Journal of Network Security*, vol. 15, no. 3, pp. 149-159, 2013.

[50] I. C. Lin, M. S. Hwang, and L. H. Li, "A New Remote User Authentication Protocol for Multi-server Architecture," *Future Generation Computer Systems*, vol. 19, no. 1, pp. 13-22, 2003.

[51] C. H. Lin and Y. Y. Lai, "A Flexible Biometric Remote User Authentication Protocol," *Computer Standards & Interfaces*, vol. 27, no. 1, pp. 19-23, 2004.

[52] C. L. Lin, H. M. Sun, M. Steiner, and T. Hwang, "Three-party Encrypted Key Exchange Without Server Public Keys," *IEEE Communications Letters*, vol. 5, no. 12, pp. 497-499, 2001.

[53] J. W. Lo, J. Z. Lee, M. S. Hwang, and Y. P. Chu, "An Advanced Password

Authenticated Key Exchange Protocol for Imbalanced Wireless Networks", *Journal of Internet Technology*, vol. 11, no. 7, pp. 997-1004, Dec. 2010.

[54] J. W. Lo, S. C. Lin, and M. S. Hwang, "A Parallel Password-authenticated Key Exchange Protocol for Wireless Environments", *Information Technology and Control*, vol. 39, no. 2, pp. 146-151, 2010.

[55] D. C. Lou and H. F. Huang, "Efficient Three-party Password-based Key Exchange Protocol," *International Journal of Communication Systems*, vol. 24, no. 4, pp. 504-512, 2011.

[56] R. Lu and Z. Cao, "Simple Three-party Key Exchange Protocol," *Computers & Security*, vol. 26, no. 1, pp. 94-97, 2007.

[57] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, "Handbook of Fingerprint Recognition," (Springer, New York, 2009, 2nd edn.)

[58] J. C. Mason and D. C. Handscomb, "Chebyshev Polynomials," *Chapman & Hall/CRC Press*, 2003.

[59] B. Menkus, "Understanding the Use of Passwords," *Computers & Security*, vol. 7, no. 2, pp. 132-136, 1988.

[60] T. Messerges, E. Dabbish, and R. Sloan, "Examining Smart-card Security Under the Threat of Power Analysis Attacks," *IEEE Transactions on Computers*, vol. 51, no. 5, pp. 541-552, 2002.

[61] H. K. Pathak and M. Sanghi, "Simple Three Party Key Exchange Protocol via Twin Diffie-Hellman Problem," *International Journal of Network Security*, vol. 15, no. 4, pp. 201-209, 2013.

[62] S. Prabhakar, S. Pankanti, and A. K. Jain, "Biometric Recognition: Security and Privacy Concerns," *IEEE Security & Privacy*, vol. 1, no. 2, pp. 33-42, 2003.

[63] J. J. Shen, C. W. Lin, and M. S. Hwang, "A Modified Remote User Authentication Protocol Using Smart Cards," *IEEE Transactions on Consumer Electronics*, vol. 49, no. 2, pp. 414-416, 2003.

[64] J. J. Shen, C. W. Lin, and M. S. Hwang, "Security Enhancement for The Timestamp-based Password Authentication Using Smart Cards," *Computers & Security*, vol. 22, no. 7, pp. 591-595, 2003.

[65] L. J. Sheu, "A Speech Encryption Using Fractional Chaotic Systems," *Nonlinear Dynamics*, vol. 65, no. 1-2, pp. 103-108, 2011.

[66] H. M. Sun, Y. H. Chen, and Y. H. Lin, "oPass: A User Authentication Protocol Resistant to Password Stealing and Password Reuse Attacks," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 651-663, 2012.

[67] D. Z. Sun, J. P. Huai, J. Z. Sun, J. X. Li, J. W. Zhang, and Z. Y. Feng, "Improvements of Juang’s Password-authenticated Key Agreement Protocol Using Smart Cards," *IEEE Transactions on Industrial Electronics*, vol. 56, no. 6, pp. 2284-2291, 2009.

[68] J. L. Tsai, "Efficient Multi-server Authentication Protocol Based on One-way Hash Function Without Verification Table," *Computers & Security*, vol. 27, no. 3-4, pp. 115-121, 2008.

[69] C. S. Tsai, C. C. Lee, and M. S. Hwang, "Password Authentication Protocols: Current Status and Key Issues," *International Journal of Network Security*, vol. 3, no. 2, pp. 101-115, 2006.

[70] W. J. Tsaur, J. H. Li, and W. B. Lee, "An Efficient and Secure Multi-server Authentication Protocol with Key Agreement," *The Journal of Systems and Software*, vol. 85, no. 4, pp. 876-882, 2012.

[71] H. R. Tseng, R. H. Jan, and W. Yang, "A Chaotic Maps-based Key Agreement

Protocol that Preserves User Anonymity," *In: IEEE International Conference on Communications, ICC'09*, pp. 1-6, 2009.

[72] B. Vaidya, J. H. Park, S. S. Yeo, and J. J. P. C. Rodrigues, "Robust One-time Password Authentication Protocol Using Smart Card for Home Network Environment," *Computer Communications*, vol. 34, no. 3, pp. 326-336, 2011.

[73] X. Wang, X. Wang, J. Zhao, and Z. Zhang, "Chaotic Encryption Algorithm Based on Alternant of Stream Cipher and Block Cipher," *Nonlinear Dynamics*, vol. 63, no. 4, pp. 587-597, 2011.

[74] Y. Wang, K. W. Wong, X. Liao, and T. Xiang, "A Block Cipher with Dynamic S-boxes Based on Tent Map," *Communications in Nonlinear Science and Numerical Simulation*, vol. 14, no. 7, pp. 3089-3099, 2009.

[75] X. Y. Wang, L. Yang, R. Liu, and A. Kadir, "A Chaotic Image Encryption Algorithm Based on Perceptron Model," *Nonlinear Dynamics*, vol. 62, no. 3, pp. 615-621, 2010.

[76] J. Wessels, "Application of BAN-Logic," *CMG Public Sector B.V.*, 2001, available at http://www.win.tue.nl/ipa/archive/springdays2001/banwessels.pdf, access date: 2013/4/22.

[77] K. W. Wong, "A Fast Chaotic Cryptographic Protocol with Dynamic Lookup Table," *Physics Letters A*, vol. 298, no. 4, pp. 238-242, 2002.

[78] S. Wu, K. Chen, and Y. Zhu, "Enhancements of a Three-party Password-based Authenticated Key Exchange Protocol," *International Arab Journal of Information Technology*, vol. 10, no. 3, May 2013.

[79] D. Xiao, X. Liao, and S. Deng, "One-way Hash Function Construction Based on the Chaotic Map with Changeable-parameter," *Chaos, Solitons & Fractals*, vol. 24, no. 1, pp. 65-71, 2005.

[80] D. Xiao, F. Shih, and X. Liao, "A Chaos-based Hash Function with Both Modification Detection and Localization Capabilities," *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, no. 9, pp. 2254-2261, 2010.

[81] Z. Yong, M. Jianfeng, and S. Moon, "An Improvement on a Three-party Password-based Key Exchange Protocol Using Weil Pairing," *International Journal of Network Security*, vol. 11, no. 1, pp. 17-22, 2010.

[82] E. J. Yoon and K. Y. Yoo, "Cryptanalysis of an Efficient Three-party Password-based Key Exchange Protocol," *Procedia Engineering*, vol. 29, pp. 3972-3979, 2012.

[83] E. J. Yoon and K. Y. Yoo, "Cryptanalysis of a Simple Three-party Password-based Key Exchange Protocol," *International Journal of Communication Systems*, vol. 24, no. 4, pp. 532-542, 2011.

[84] E. J. Yoon and K. Y. Yoo, "Cryptanalysis of Robust E-mail Protocols with Perfect Forward Secrecy," *IEEE Communication Letters*, vol. 11, no. 5, pp. 372-374, 2007.

[85] W. Yuan, L. Hu, H. Li, and J. Chu, "Offline Dictionary Attack on a Universally Composable Three-party Password-based Key Exchange Protocol," *Procedia Engineering*, vol. 15, pp. 1691-1694, 2011.

[86] L. Zhang, "Cryptanalysis of the Public Key Encryption Based on Multiple Chaotic Systems," *Chaos, Solitons & Fractals*, vol. 37, no. 3, pp. 669-674, 2008.