南　　華　　大　　學

資訊管理學系

碩士論文

多 重 伺 服 器 通 行 碼 認 證 協 定 之 安 全 性 研 究

Security weaknesses in two multi-server password based authentication protocols

研 究 生：丁振中

指導教授：周志賢 博士

中華民國九十八年六月

南　華　大　學

資訊管理學系

碩 士 學 位 論 文

多重伺服器通行碼認證協定之安全性研究

Security weaknesses in two multi-server password based authentication protocol

研究生：丁振中

經考試合格特此證明

口試委員：

周志賢

指導教授：周志賢

系主任(所長)：

口試日期：中華民國九十八年六月二十四日

誌　　　　謝

又是驪歌聲響起，回想兩年來碩士班修業種種，是人生難得的寶貴學習經驗，本論文得以完成，首先感謝指導教授，我的恩師　周志賢博士悉心教導與關懷，在此謹致上由衷敬意及謝忱。

感謝口試委員　許乙清博士與　尤國任博士對本論文提供許多建設性的意見與改進的方向，使本論文更臻完整，謹此致謝。

最後，感謝我的家人及陪我一起走過研究期間的同學及好友，你們給予的鼓勵和支持，讓我有了繼續前進的原動力，在求學路上充滿溫馨。

丁振中 謹誌

多重伺服器通行碼認證協定之安全性研究

學生：丁振中　　　　　　　　　　　指導教授：周志賢 博士

南　華　大　學　資訊管理學系碩士班

摘　　　要

在 2004 年及 2005 年， Tsaur 等人分別提出二個運用智慧卡在網際網路遠端環境建立多重伺服器通行碼認證通信協定。他們聲稱他們的協定是安全的，並且可能承受各種各樣的種類攻擊。然而，在分析以後，我們發現他們的每一個協定之中都存在一些安全性的缺失。在本文中，我們將揭示這二個協定的安全缺失。

關鍵詞： 多重伺服器，遠程通行碼認證，智慧卡，金鑰協議，
拉格朗日插補多項式

Security weaknesses in two multi-server password based authentication protocols

Student：Cheng-Chung Ding Advisors：Dr. Jue-Sam Chou

Department of Information Management
The M.I.M. Program
Nan-Hua University

ABSTRACT

In 2004 and 2005, Tsaur et al. proposed a smart card based password authentication schemes for multi-server environments, respectively. They claimed that their protocols are safe and can withstand various kinds of attacks. However, after analysis, we found their schemes each have some secure loopholes. In this article, we will show the security flaws in these two protocols.

Keywords: multi-server, remote password authenticationl, smart card, key agreement, lagrange interpolating polynomial.

目　錄

圖　目　錄

Chapter 1 Introduction

In 1974, Roland Moreno invented integrated circuits card (IC card or smart card). At that time, it was used as the debit card by the bank. Recenlly, the smart card applications have got rapid progress not only for its promotion in the aspects of security and processing speed but also for its significant reduction in cost. It has widely accepted as an important tool in human's life for its capablity of achieving the goals of integrity, privacy, authentication, etc.

In a traditional identity authentication mechanism, the user must first use his identity ID and password PW to registrate at the remote server. The remote server then establishes the verification table for recording this pair of ID and PW. Thereafter, when the legitimate user wants to login the system, he must first transmit his ID and the protected PW to the remote server. The server then looks up verification table to see whether or not the user has existed in the table. If it has, the server consider that the user is valid. He then provides the required resources to the uesr. However this framework is too simple to be secure. It is easy to suffer from a passive or active attacker over the Internet. In 1981, Lamport [9] proposed a remote password authentication scheme. It emphasizes that it can prevent the replay attack. However, it needs to establish a password verification table in the remote server to authenticate the user. Although, it uses an one-way hash function to protect the password, the attacker still might be able to find out the password for the fact that the password itself is weak or it may suffer the stolen verifier attack. To address this problem, some researchers proposed methods for authenticating the remote user by using the non-verification table way.

In a client-server protocol, the client has to regist to the server. Then, he can login to the server for accessing the server's resources by typing the password for the corresponding server. In this environment, he has to remember different passwords for different server which he registers at. In a multi-server protocol, the client can remember only one password to access resources of all servers in the system if he registers at the registration center which manages about who can access the system. Each of protocols [3, 8, 13-15] is a multi-server protocol with client's password or protected password stored in smart card. A smart card is a plastic card embedded with a memory chip and a microprocessor which can process data. It is easy to be carried, can store information, and performs calculations. In a consequence, for security consideration, most protocols adopt smart cards storing passwords to authenticate clients.

Generally speaking, a multi-server protocol consists of four phases. They are the preparation phase, the registration phase, the login phase, and the authentication phase. The preparation phase is that every members, clients and servers, register to the registration center for preparing needed parameters in the system. The login phase is that when a user wants to access a server's resource, he starts the protocol and sends a message to the server for logging. After receiving the login message, the server and the the client performs the authentication phase to see if each other is valid. Meanwhile, they negotiate a session key for secure communications.

A secure and efficient multi-server authentication protocol should meet the following six requirements [8]. (1) Each server stores no verification table. (2) Users can freely choose and change their passwords. (3) The protocol are low computation and communication cost. (4) The protocol makes a server and a user achieve mutual authentication. (5) A server and a user negotiate a session key to protect their subsequent communications. (6) Users register at the register center once and can use all servers' resource. In addition, the protocol should meet one more requirement as indicated in [3] : the protocol can resist all kinds of attacks.

In 1990, Hwang et al. [6] first proposed a smart card based non-verification table mechanism for authentication. Thereafter, many schemes [1, 2, 4, 7, 10, 13] were proposed based on this non-verification-table type. These authentication mechanisms protect the transmitted information either by the discrete logarithm problem (DLP) or by the asymmetric encryption method. In 2004 and 2005, Tsaur et al. proposed two smart card based password authentication schemes [14, 15] for multi-server environments based on the non-verification-table type. They took the RSA asymmetric encryption and Lagrange interpolating polynomial as the foundation of the research. They claimed that their scheme is safe and can withstand various kinds of attacks. However, after analysis, we found that their schemes each have some security loopholes. In this article, we will demonstrate the security flaws.

The rest of this paper is organized as follows. In Section 2, we describe the background concepts of RSA cryptosystem and some related concept of mathematical problems. In Section 3, we review and show the attacks on Tsaur et al.'s two protocols. Finally, a conclusion is given in Section 4.

Chapter 2 Background concepts

In this section, we briefly review the basic concept of RSA cryptosystem [11], threshold scheme, and lagrange interpolating polynomial.

2.1 RSA cryptosystem

Since 1976, Diffie and Hellman proposed the concept of public key cryptography (PKC) [5], a new era of cryptology research has been opened. PKC belongs to the asymmetric cryptographic system. In this type of encryption, whenever a sender wants to transmit information to the receiver, he uses the receiver's public to encrypt the information. Conversely, when the receiver receives the message from the sender, he uses his private key to decrypt the encryption, obtaining the plaintext of the information. It is infeasible for an attacker to obtain the receiver's private key only with using the receiver's public key and some public information in the system. Although, it is time-consuming in the encryption and decryption computation process for its using the modular exponentiation operations, it is suitable for short message encryption, e.g., the session key encryption, and can be applied in many situations such as signing and key exchange. Its security is based on the difficulty of factorization. The factorization problem is now still a NP-complete problem.

2.2 Threshold scheme and lagrange interpolating polynomial polynomial

In 1979, Shamir [12] proposed the first (*t, n*) threshold secret sharing scheme based on lagrange interpolating polynomial. In it, a secret *K* can be shared among n participants. The secret dealer must distribute every participant's a secret shadow. Only at least *t* or more participants can reconstruct the secret *K*. Conversely, if the number of participants is less than *t*, the participants can obtain nothing about the secret *K*. This method is mainly used in a plane containing *t* points to decide the polynomial with degree *t*-1. Takeing *t* as the threshold value and appling the Lagrange interpolating polynomial, we can obtain the polynomial. In the Following, we roughly describe the formation of the polynomial:

- The dealer chooses a secret *K* and a prime number *p* which and satisfies $p \geqq K$.
- The dealer randomly chooses *t-1* degree polynomial of $F(x) = a_{t-1}x^{t-1} + a_{t-2}x^{t-2} + \ldots + a_2x_2 + a_1x_1 + K \pmod p$, where $a_{t-1}, a_{t-2}, \ldots, a_2,$ and a_1 are all random integer, with rang in [1, *p*-1].
- Let each participant's identity be x_i, $1 \le i \le n$. The dealer rests on x_i deduce the subkey $y_i = F(x_i)$ for each participant.
- When the number of subkeys is greater than *t*, they (the participants) can contruct the polynomial to obtain the shared secret *K* by letting $x_i = 0$. In the following, we roughly describe the Lagrange interpolating polynomial.

Let $\{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}$ be *n* distinct points. Then, the *t*-1 degree polynomial *F(x)* can be *F* formed by the following formula.

$$F(x) = y_1 \frac{(x - x_2)(x - x_3)\ldots(x - x_n)}{(x_1 - x_2)(x_1 - x_3)\ldots(x_1 - x_n)} + y_2 \frac{(x - x_1)(x - x_3)\ldots(x - x_n)}{(x_2 - x_1)(x_2 - x_3)\ldots(x_2 - x_n)}$$

$$+\ldots\ldots+y_n\frac{(x-x_1)(x-x_2)\ldots(x-x_{n-1})}{(x_n-x_1)(x_n-x_2)\ldots(x_n-x_{n-1})}$$

$$=\sum_{i=1}^{n}y_i\prod_{j=1,j\neq i}^{n}\frac{(x-x_j)}{(x_i-x_j)}$$

Chapter 3 Review and attack on two Tsaur et al.'s protocols

In this section, we will review and attack on Tsaur et al.'s first protocol in Section 3.1 and on second protocol in Section 3.2. Before that, the notations used throughout this paper are first defined as follows.

CA : the central authority

S_j : a legal server j

U_i : a legal user i

ATT_e : a malicious attacker

p_1, p_2 : two distinct large primes

N, P, e : CA's public keys

d : CA's secret key

S_SK_j : the secret key of S_j

S_ID_j : the identity of S_j

E_T_{ij} : S_j's service period for U_i

U_ID_i : the identity of U_i

U_PW_i : the password of U_i

U_R_i, U_S_i : U_i's two secret keys

U_SC_i : U_i's smart card

$f_i(X)$: a lagrange interpolating polynomial that CA constructs for U_i

M : an authentication message

$h(X,Y)$: an one-way hash function with two parameters X and Y

g : a primitive element in a Galois field GF(p), where p is a large prime number

t : a timestamp

ΔT : endurable transmission delay time

=> : a secure channel

→ : a common channel

3.1 Review and attack on Tsaur et al.'s first protocol

(A) Review of Tsaur et al.'s first protocol

Tsaur et al.'s first protocol [14] consists of four stages. They are: (1)The system setup stage, (2)The user registration stage, (3)The log-in stage, and (4)The server authentication stage. We depict their scheme in figure 1 and also describe it as follows.

(1) The system setup stage

In this phase, CA selects the server's private key and computes its identity. CA's operations are described as follows:

- According to RSA cryptographic algorithm, CA first selects two large prime numbers p_1, p_2, computes $N = p_1 \times p_2$, randomly chooses the encryption key e satisfying gcd(e, $\phi(N)$) = 1, where $\phi(N) = (p_1 - 1) \times (p_2 - 1)$ as his public key, and then uses the Extended Euclean Algorithm to compute his corresponding private key as $d = e^{-1} \bmod \phi(N)$.
- For each server S_j, CA selects S_SK_j and computes $S\_ID_j = g^{S\_SK_j} \pmod N$ as S_j's

private key and identity respectively, where j = 1,2, ..., m.

- In addition, it also chooses an one-way hash function *h(X,Y)* for the system.

(2) The user registration stage

Assume that a new user U_i wants to register at m servers S_1, S_2, …, and S_m in a multi-server system. The entire registration process is described as follows (also shown in Fig. 1):

- U_i chooses his identity U_ID_i and password U_PW_i and transmits them to CA.
- CA randomly chooses a number r_{ui} for U_i, and computes U_i's two secret keys as follows:

  $U\_R_i = g^{U\_PW_i * r_{ui}} \pmod N$

  $U\_S_i = g^{r_{ui} * d} \pmod N$

- CA assumes that Ui wants to obtain server S_j's service, $1 \le j \le r < m$. The service periods provided by the servers for U_i are E_T_{i1}, E_T_{i2}, …, and E_T_{ir} respectively. The other periods for the other servers S_{r+1}, S_{r+2}, …, and S_m are all set to zeros. CA then constructs a Lagrange interpolating polynomial function $f_i(X)$ for Ui as follows:

$$f_i(X) = \sum_{j=1}^{m}(U\_ID_i + E\_T_{ij})\frac{(X - U\_ID_i)}{(S\_SK_j - U\_ID_i)} \times \prod_{k=1,k\neq j}^{m}\frac{(X - S\_SK_k)}{(S\_SK_j - S\_SK_k)}$$

$$+ U\_R_i \prod_{y=1}^{m}\frac{(X - S\_SK_y)}{(U\_ID_i - S\_SK_y)} \pmod N$$

$$= a_m X^m + a_{m-1}X^{m-1} + \ldots + a_1 X + a_0 \pmod N$$

- CA stores $f_i(X)$, U_i's identity U_ID_i, secret keys U_S_i *and* U_R_i, and the one-way function $h(X,Y)$ in U_i's smart card U_SC_i. Then CA sends the card to U_i via a secure channel.

(3) The log-in stage

In this phase, when a registered user U_i wants to login to server S_j, it inserts his smart card U_SC_i to the reader and keys in his U_PW_i. Then, U_i performs following steps by using U_SC_i:

- U_SC_i gets a timestamp t from the system. Then, it generates a secret random number r_1, and computes C_1, C_2, and P as follows:

$$C_1 = g^{e*r_1} \pmod N$$

$$C_2 = (U\_S_1)^{U\_PW_i} \cdot g^{r_1 * h(C_1, t)}$$

$$= g^{U\_PW_i * r_{ui} * d} \cdot g^{r_1 * h(C_1, t)} \pmod N$$

$$P = (S\_ID_j)^{e*r_1} \pmod N = (g^{S\_SK_j})^{e*r_1} \pmod N = g^{S\_SK_j * e * r_1} \pmod N$$

- Given 1, 2, …, m, and P, U_SC_i computes $f_i(1)$, $f_i(2)$, …, $f_i(m)$, and $f_i(P)$. Then, it constructs an authentication message $M = \{U\_ID_i, t, C_1, C_2, f_i(1), f_i(2), \ldots, f_i(m), f_i(P)\}$ and sends it to S_j, one of the m servers for, $1 \le j \le m$.

(4) The server authentication stage

In this phase, after receiving the authentication message from U_i, S_j requires his system to obtain current timestamp t_{now} and performs the following steps to verify the login message from U_i:

- Checks U_i's identity U_ID_i and whether or not $t_{now} - t > \Delta T$, if U_i's identity U_ID_i is invalid or $t_{now} - t > \Delta T$, S_j rejects; otherwise, it continues.
- It uses the value C_1 and its secret key S_SK_j to derive the value P shown as below.

$$P = (C_1)^{S\_SK_j} \pmod N = (g^{e*r_1})^{S\_SK_j} \pmod N = g^{e*r_1*S\_SK_j} \pmod N$$

Then, it uses these m + 1 points $\{(1, f_i(1)), (2, f_i(2)), \ldots, (m, f_i(m)), (P, f_i(P))\}$ to reconstruct the interpolating polynomial

$$f_i(X) = a_m X^m + a_{m-1} X^{m-1} + \ldots + a_1 X + a_0 \pmod N.$$

- Checks to see whether $\dfrac{(C_2)^e}{(C_1)^{h(C_1,t)} \cdot U\_R_i} = 1$, if it holds, user U_i is qualified.

Otherwise, U_i is rejected. The verification formula is shown as follows.

$$\frac{(C_2)^e}{(C_1)^{h(C_1,t)} \cdot U\_R_i} = \frac{(g^{U\_PW_i*r_{ui}*d} \cdot g^{r_1*h(C_1,t)})^e}{g^{e*r_1*h(C_1,t)} \cdot g^{U\_PW_i*r_{ui}}}$$

$$= \frac{g^{U\_PW_i*r_{ui}} \cdot g^{r_1*h(C_1,t)*e}}{g^{e*r_1*h(C_1,t)} \cdot g^{U\_PW_i*r_{ui}}} = 1 \pmod N$$

The system setup stage

CA

1. chooses prime numbers p_1 and p_2
   computes $N = p_1 \times p_2$
   computes $\phi(N) = (p_1 - 1) * (p_2 - 1)$
   chooses random public key e satisfying gcd(e, $\phi(N)$) = 1
   computes secret key $d = e^{-1} \bmod \phi(N)$
2. (For each server S_j, j=1, 2, ..., m):
   chooses private key S_SK_j
   computes S_j's identity as $S\_ID_j = g^{S\_SK_j} \pmod N$
3. chooses one-way hash function $h(X,Y)$

The user registration stage

U_i | CA

1. chooses U_ID_i, U_PW_i

$\xrightarrow{U\_ID_i,\ U\_PW_i}$

2. chooses a random number r_{ui}
   computes $U\_R_i = g^{U\_PW_i * r_{ui}} \pmod N$
   computes $U\_S_i = g^{r_{ui} * d} \pmod N$

   assumes that Ui wants to obtain the service of server S_i, $1 \le i \le r < m$. The periods which the r servers provide for U_i are E_T_{i1}, E_T_{i2}, …, and E_T_{ir} respectively. The other periods for the other servers S_{r+1}, S_{r+2}, …, and S_m are all set to zeros.
   constructs

$$f_i(X) = \sum_{j=1}^{m} (U\_ID_i + E\_T_{ij}) \frac{(X - U\_ID_i)}{(S\_SK_j - U\_ID_i)}$$

$$\times \prod_{k=1, k \ne j}^{m} \frac{(X - S\_SK_k)}{(S\_SK_j - S\_SK_k)}$$

$$+ U\_R_i \prod_{y=1}^{m} \frac{(X - S\_SK_y)}{(U\_ID_i - S\_SK_y)} \pmod N$$

$$= a_m X^m + a_{m-1} X^{m-1} + \ldots + a_1 X + a_0 \pmod N$$

3. stores $f_i(X)$, U_ID_i, U_S_i, and an one-way function $h(X,Y)$ to U_i's smart card U_SC_i

$\xleftarrow{U\_SC_i}$

Fig. 1. Review of Tsaur et al.'s first protocol

The log-in stage

U_i S_j

1. inserts smart card U_SC_i to a reader
   keys in U_PW_i

U_SC_i

1. gets a timestamp t
   generates a secret random number r_1
   computes

$$C_1 = g^{e*r_1} \pmod N$$

$$C_2 = (U\_S_1)^{U\_PW_i} \cdot g^{r_1 * h(C_1,t)}$$

$$= g^{U\_PW_i * r_{ui} * d} \cdot g^{r_1 * h(C_1,t)} \pmod N$$

$$P = (S\_ID_j)^{e*r_1} \pmod N$$
$$= (g^{S\_SK_j})^{e*r_1} \pmod N$$
$$= g^{S\_SK_j * e * r_1} \pmod N$$

2. given 1, 2, …, m, and P,
   computes $f_i(1), f_i(2), \ldots, f_i(m)$, and $f_i(P)$
3. constructs an authentication message:
   $M = \{U\_ID_i, t, C_1, C_2, f_i(1), f_i(2), \ldots, f_i(m), f_i(P)\}$

$$\xrightarrow{\quad M \quad}$$

The server authentication stage

S_j

1. t_{now} is current timestamp
   checks U_i's identity U_ID_i and whether or not $t_{now} - t > \Delta T$
   if U_ID_i is invalid or $t_{now} - t > \Delta T$, *rejects*; otherwise, continues
2. computes

$$P = (C_1)^{S\_SK_j} \pmod N = (g^{e*r_1})^{S\_SK_j} \pmod N$$
$$= g^{e*r_1*S\_SK_j} \pmod N$$

3 uses points $\{(1, f_i(1)), (2, f_i(2)), \ldots, (m, f_i(m)), (P, f_i(P))\}$ to reconstruct .

$$f_i(X) = a_m X^m + a_{m-1} X^{m-1} + \ldots + a_1 X + a_0 \pmod N$$

4. Verifies whether or not $\dfrac{(C_2)^e}{(C_1)^{h(C_1,t)} \cdot U\_R_i} = 1$,
   if it holds, user U_i is qualified. Otherwise, U_i is rejected.

Fig. 1-continued Review of Tsaur et al.'s first protocol

(B) Attack on Tsaur et al.'s first protocol

Tsaur et al. claimed that their protocol is safe and can withstand various kinds of attacks. In this section, we will show that their protocol is vulnerable (as shown in Fig. 2). In the following, we will describe that there exists a weakness in Tsaur et al.'s first protocol. Since that a malicious adversary ATT_e can successfully launch an attack shown as follows:

(1) Assume that there is a malicious attacker ATT_e who wants to disguise as user U_i, a legal user in the system, to login to S_j. Before the login stage, ATT_e purchases a smart card and pretends to be CA by preparing the needed parameters stored in the card for the login stage. ATT_e performs as follows.

- Enters *U_ID*$_i$, randomly chooses a password *U_PW*$_i$, selects a number r_{ui}, and calculates U_i's two secrets as follows.

  $U\_R_i = g^{U\_PW_i * r_{ui} * e} \pmod N$

  $U\_S_i = g^{r_{ui}} \pmod N$

- Then, it acts as CA. Though, ATT_e does not know each server's private key, it knows these servers' identities. Therefore, it can use each server's identity to replace the original corresponding private key in the computation of $f_i(X)$ as shown in Equation (1).

$$f_i(X) = \sum_{j=1}^{m}(U\_ID_i + E\_T_{ij})\frac{(X - U\_ID_i)}{(S\_ID_j - U\_ID_i)} \times \prod_{k=1,k\neq j}^{m}\frac{(X - S\_ID_k)}{(S\_ID_j - S\_ID_k)}$$

$$+ U\_R_i \prod_{y=1}^{m}\frac{(X - S\_ID_y)}{(U\_ID_i - S\_ID_y)} \pmod N$$

$$= a_m X^m + a_{m-1}X^{m-1} + \ldots + a_1 X + a_0 \pmod N \qquad \cdots\cdots\cdots\cdots \text{Equation (1)}$$

(2) In log-in stage, when ATT_e wants to login to server S_j, It performs the follows steps:

- ATT_e gets a timestamp t from the system. Then it generates a secret random number r_1', and computes C_1, C_2, and P as follows:

  $C_1 = g^{e * r_1'} \pmod{N}$,

  $C_2 = (U\_S_1)^{U\_PW_i} \cdot g^{r_1' * h(C_1, t)}$

  $\quad = g^{U\_PW_i * r_{ui}} \cdot g^{r_1' * h(C_1, t)} \pmod{N}$,

  $P = (S\_ID_j)^{e * r_1'} \pmod{N} = (g^{S\_SK_j})^{e * r_1'} \pmod{N} = g^{S\_SK_j * e * r_1'} \pmod{N}$.

- Then, ATT_e computes $f_i(1)$, $f_i(2)$, …, $f_i(m)$, and $f_i(P)$ and sends an authentication message $M = \{U\_ID_i, t, C_1, C_2, f_i(1), f_i(2), \ldots, f_i(m), f_i(P)\}$ to server S_j.

(3) The server authentication stage

When receiving the authentication message from ATT_e, S_j records the current timestamp in t_{now}. He then performs the following verification steps to authenticate ATTe.

- checks ATT_e's identity U_ID_i and whether or not $t_{now} - t > \Delta T$, if the identity U_ID_i is invalid or $t_{now} - t > \Delta T$, S_j rejects; otherwise, it continues.
- S_j uses the transmitted value C_1 and his secret key S_SK_j to derive the value P (shown as below in Equation (2)),

  $P = (C_1)^{S\_SK_j} \pmod{N} = (g^{e * r_1'})^{S\_SK_j} \pmod{N}$

  $\quad = g^{e * r_1' * S\_SK_j} \pmod{N}$ ……………Equation (2),

  then uses these m + 1 points $\{(1, f_i(1)), (2, f_i(2)), \ldots, (m, f_i(m)), (P, f_i(P))\}$ to reconstruct the interpolating polynomial

  $f_i(X) = a_m X^m + a_{m-1} X^{m-1} + \ldots + a_1 X + a_0 \pmod{N}$.

- S_j verifies whether or not $\frac{(C_2)^e}{(C_1)^{h(C_1,t)} \cdot U\_R_i} = 1$, if it holds, ATT_e is authentic .

  Obviously, ATT_e can pretend as U_i successfully since the computation result is equal to 1 as shown below in Equation (3).

$$\frac{(C_2)^e}{(C_1)^{h(C_1,t)} \cdot U\_R_i} = \frac{(g^{U\_PW_i * r_{ui}} \cdot g^{r_1' * h(C_1,t)})^e}{g^{e * r_1' * h(C_1,t)} \cdot g^{U\_PW_i * r_{ui} * e}}$$

$$= \frac{g^{U\_PW_i * r_{ui} * e} \cdot g^{r_1' * h(C_1,t) * e}}{g^{e * r_1' * h(C_1,t)} \cdot g^{U\_PW_i * r_{ui} * e}}$$

$$= 1 \pmod{N} \qquad \cdots\cdots\cdots\cdots\cdots\cdots \text{Equation (3)}$$

The system setup stage

CA

1. chooses prime numbers p_1 and p_2
   computes $N = p_1 \times p_2$
   computes $\phi(N) = (p_1 - 1) * (p_2 - 1)$
   chooses random public key e satisfying $\gcd(e, \phi(N)) = 1$
   computes secret key $d = e^{-1} \bmod \phi(N)$
2. (For each server S_j, j=1, 2, ..., m):
   chooses private key S_SK_j
   computes S_j's identity $S\_ID_j = g^{S\_SK_j} \pmod N$
3. chooses one-way hash function $h(X,Y)$

The user registration stage

ATT_e

1. chooses U_ID_i, U_PW_i
2. chooses random number r_{ui}
   computes $U\_R_i = g^{U\_PW_i * r_{ui} * e} \pmod N$
   computes $U\_S_i = g^{r_{ui}} \pmod N$
   uses S_j's identity to replace the original corresponding private key
   constructs

$$f_i(X) = \sum_{j=1}^{m} (U\_ID_i + E\_T_{ij}) \frac{(X - U\_ID_i)}{(S\_ID_j - U\_ID_i)}$$

$$\times \prod_{k=1, k \neq j}^{m} \frac{(X - S\_ID_k)}{(S\_ID_j - S\_ID_k)}$$

$$+ U\_R_i \prod_{y=1}^{m} \frac{(X - S\_SK_y)}{(U\_ID_i - S\_SK_y)} \pmod N$$

$$= a_m X^m + a_{m-1} X^{m-1} + \ldots + a_1 X + a_0 \pmod N$$

3. stores $f_i(X)$, U_ID_i, U_S_i, and an one-way function $h(X,Y)$ to ATT_e's storage device

The log-in stage

ATT_e | S_j

1. keys in U_PW_i
2. gets a timestamp t
   generates a secret random number r_1'
   computes

$$C_1 = g^{e * r_1'} \pmod N$$

Fig. 2. Attack on Tsaur et al.'s first protocol

$$C_2 = (U\_S_1)^{U\_PW_i} \cdot g^{r_1' * h(C_1, t)}$$
$$= g^{U\_PW_i * r_{ui} * d} \cdot g^{r_1' * h(C_1, t)} \pmod N$$

$$P = (S\_ID_j)^{e * r_1'} \pmod N$$
$$= (g^{S\_SK_j})^{e * r_1'} \pmod N$$
$$= g^{S\_SK_j * e * r_1'} \pmod N$$

3. computes $f_i(1), f_i(2), \ldots, f_i(m)$, and $f_i(P)$

4. constructs an authentication message:

$$M = \{U\_ID_i, t, C_1, C_2, f_i(1), f_i(2), \ldots, f_i(m), f_i(P)\}$$

$$\xrightarrow{\quad M \quad}$$

The server authentication stage

S_j

1. t_{now} is current timestamp, checks the validity of identity U_ID_i and whether or not $t_{now} - t > \Delta T$ if U_ID_i is invalid or $t_{now} - t > \Delta T$, S_j *rejects* ; otherwise, it continues

2. computes

$$P = (C_1)^{S\_SK_j} \pmod N = (g^{e * r_1'})^{S\_SK_j} \pmod N$$
$$= g^{e * r_1' * S\_SK_j} \pmod N$$

3 uses points $\{(1, f_i(1)), (2, f_i(2)), \ldots, (m, f_i(m)), (P, f_i(P))\}$ to reconstruct

$$f_i(X) = a_m X^m + a_{m-1} X^{m-1} + \ldots + a_1 X + a_0 \pmod N$$

4. checks whether $\frac{(C_2)^e}{(C_1)^{h(C_1, t)} \cdot U\_R_i} = 1$ or not ,

if it holds, ATT_e is verified. ATT_e can pretend as U_i successfully.

Fig. 2-continued Attack on Tsaur et al.'s first protocol

3.2 Review and attack on Tsaur et al.'s second protocol

(A) Review of Tsaur et al.'s second protocol

Tsaur et al.'s second protocol [15] consists of four stages. They are: (1)The system setup stage, (2)The user registration stage, (3)The login stage, and (4)The server authentication stage. We describe them as follows and also depict it in Fig.3.

(1) The system setup stage

The CA selects a large number *P*, publishes a generator g of Z_P^*, and an one-way hash function *h(X,Y),* then it selects a secret key S_SK_j for server S_j and computes S_j's identity as $S\_ID_j = g^{S\_SK_j}$ (mod *P*), $1 \leq j \leq m$.

(2)The user registration stage

In this phase, assume that a new user U_i wants to register at the m servers S_1, S_2, …, and S_m in a multi-server system. The entire registration process is described as follows (also shown in Fig. 3):

- U_i chooses his identity U_ID_i and password U_PW_i and transmits them to CA.
- CA randomly chooses a number *r*, larger than 160 bits for U_i, and computes U_i's two secret keys as follows:

  $U\_R_i = g^r \text{ (mod } P)$

  $U\_S_i = r^{-U\_PW_i} \text{ (mod } P)$

- CA supposes that U_i wants to obtain the service of one server S_i among all of the servers, $1 \leq i \leq r < m$. Assume that the service periods which serves for U_i is E_T_{i1}, E_T_{i2}, …, and E_T_{ir} respectively. The other periods for the other servers S_{r+1}, S_{r+2}, …, and S_m are all set to zeros. CA then uses S_j's secret key S_SK_j to construct a Lagrange interpolating polynomial function $f_i(X)$ for U_i as follows:

$$f_i(X) = \sum_{j=1}^{m}(U\_ID_i + E\_T_{ij})\frac{(X - U\_ID_i)}{(S\_SK_j - U\_ID_i)} \times \prod_{k=1,k\neq j}^{m}\frac{(X - S\_SK_k)}{(S\_SK_j - S\_SK_k)}$$

$$+ U\_R_i \prod_{y=1}^{m}\frac{(X - S\_SK_y)}{(U\_ID_i - S\_SK_y)} \pmod N$$

$$= a_m X^m + a_{m-1}X^{m-1} + \ldots + a_1 X + a_0 \pmod N$$

- CA then stores U_S_i and $f_i(X)$ in U_i's smart card U_SC_i secret data space, and sends it to U_i via a secure channel.

(3)The login stage

In this phase, when a registered user U_i wants to login to server S_j, it inserts his smart card U_SC_i to the reader and keys in his U_PW_i. Then, U_i performs the following steps by using U_SC_i:

- U_SC_i gets a timestamp t from the system, and computes $r = (U\_Si)^{U\_PW_i}$. Then, it generates a secret random number r_1 and computes C_1, C_2 and p *as* follows.

  $$C_1 = g^{r_1} \pmod P$$

  $$C_2 = r_1 + r \cdot h(C_1, t) \pmod P$$

  $$p = (S\_ID_j)^{r_1} \pmod P$$

- Given 1, 2, …, m, and p, U_SC_i computes $f_i(1)$, $f_i(2)$, …, $f_i(m)$, and $f_i(p)$. Then, it constructs an authentication message $M = \{U\_ID_i, t, C_1, C_2, f_i(1), f_i(2), \ldots, f_i(m), f_i(p)\}$ and sends it to S_j, $1 \leq j \leq m$.

(4)The server authentication stage

In this phase, When S_j receives the authentication message from U_i, S_j obtains a current timestamp t_{now} from his system and performs the following steps to verify the login message from U_i:

- Checks U_i's identity U_ID_i and whether or not t_{now} - $t > \Delta T$. If both hold, Sj computes

  $p = (C_1)^{S\_SK_j}$ (mod P)

- uses the received m + 1 points $\{(1, f_i(1)), (2, f_i(2)), \ldots, (m, f_i(m)), (P, f_i(P))\}$ from U_ID_i to reconstruct the interpolating polynomial

  $f_i(X) = a_m X^m + a_{m-1} X^{m-1} + \ldots + a_1 X + a_0 \pmod{N}$.

- Checks to see whether $\frac{g^{C_2}}{(C_1) \cdot (U\_R_i)^{h(C_1, t)}} = 1$, if it holds, user U_i is qualified. Otherwise, U_i is rejected. The verification formula is shown as follows.

$$\frac{g^{C_2}}{(C_1) \cdot (U\_R_i)^{h(C_1, t)}} = \frac{g^{r_1 + r * h(C_1, t)}}{g^{r_1} \cdot g^{r * h(C_1, t)}}$$

$$= \frac{g^{r_1 + r * h(C_1, t)}}{g^{r_1 + r * h(C_1, t)}}$$

$$= 1 \pmod{P}$$

The system setup stage

CA

1. chooses prime numbers P

chooses an one-way hash function $h(X,Y)$

$g \in Z_P^*$

2. (For each server S_j, $1 \leq j \leq m$):

chooses secret key S_SK_j

computes S_j's identity $S\_ID_j = g^{S\_SK_j} \pmod P$

The user registration stage

U_i | CA

1. chooses U_ID_i, U_PW_i

$$\xrightarrow{U\_ID_i,\ U\_PW_i}$$

2. chooses random number r

computes $U\_R_i = g^r \pmod P$

computes $U\_S_i = r^{-U\_PW_i} \pmod P$

assumes that Ui wants to obtain the servers S_i, $1 \leq i \leq r < m$, service. The service periods which servers serve for U_i are E_T_{i1}, E_T_{i2}, …, and E_T_{ir} respectively, the other periods for the other servers S_{r+1}, S_{r+2}, …, and S_m are all set to zeros.

constructs

$$f_i(X) = \sum_{j=1}^{m}(U\_ID_i + E\_T_{ij})\frac{(X - U\_ID_i)}{(S\_SK_j - U\_ID_i)} \times \prod_{k=1,k\neq j}^{m}\frac{(X - S\_SK_k)}{(S\_SK_j - S\_SK_k)} + U\_R_i\prod_{y=1}^{m}\frac{(X - S\_SK_y)}{(U\_ID_i - S\_SK_y)} \pmod P$$

$$= a_m X^m + a_{m-1}X^{m-1} + \ldots + a_1 X + a_0 \pmod P$$

3.stores $U\_S_i, f_i(X)$ to U_i's smart card U_SC_i

$$\xleftarrow{U\_SC_i}$$

The login stage

U_i | S_j

1. inserts smart card U_SC_i to a reader

keys in U_PW_i

Fig. 3. Review of Tsaur et al.'s second protocol

$\boxed{U\_SC_i}$

1. gets a timestamp t
   generates a secret random number r_1
   computes

$$C_1 = g^{r_1} \pmod{P}$$

$$C_2 = r_1 + r \cdot h(C_1,\ t) \pmod{P}$$

$$p = (S\_ID_j)^{r_1} \pmod{P}$$

2. given 1, 2, …, m, and p;
   computes $f_i(1), f_i(2), \ldots, f_i(m)$, and $f_i(p)$
3. constructs an authentication message:
   $M = \{U\_ID_i, t, C_1, C_2, f_i(1), f_i(2), \ldots, f_i(m), f_i(p)\}$

$$\xrightarrow{\quad M \quad}$$

The server authentication stage

$\boxed{S_j}$

1. t_{now} is current timestamp
   checks U_i's identity U_ID_i and whether or not $t_{now} - t > \Delta T$
   if U_ID_i is invalid or $t_{now} - t > \Delta T$, *rejects*; otherwise, continues
2. computes

$$p = (C_1)^{S\_SK_j} \pmod{P}$$

3. uses points $\{(1, f_i(1)), (2, f_i(2)), \ldots, (m, f_i(m)), (p, f_i(p))\}$ to reconstruct

$$f_i(X) = a_m X^m + a_{m-1} X^{m-1} + \ldots + a_1 X + a_0 \pmod{P}$$

4. Verifies whether or not $\dfrac{g^{C_2}}{(C_1)\cdot(U\_R_i)^{h(C_1,t)}} = 1$,
   if it holds, user Ui is qualified. Otherwise, U_i is rejected.

Fig. 3-continued Review of Tsaur et al.'s second protocol

(B) Attack on Tsaur et al.'s second protocol

Tsaur et al. claimed that their protocol is safe and can withstand various kinds of attacks. In this section, we will show that their protocol is vulnerable (as shown in Fig.4). In the following, we will describe that there exists a weakness in Tsaur et al.'s second protocol. Since that a malicious adversary ATT_e can successfully launch an attack shown as follows.

(1) Assume that there is a malicious attacker ATT_e wants to disguise as user U_i, who is a legal user recorded in the system, to login to S_j. Before the login stage, ATT_e purchases a smart card and pretends to be CA to prepare the needed parameters for being stored in his card for the login stage. ATT_e performs as follows.

- Enters U_ID_i, randomly chooses a password U_PW_i and a number r larger than 160 bits, and computes U_i's two secrets as follows.

  $U\_R_i = g^r \pmod P$

  $U\_S_i = r^{-U\_PW_i} \pmod P$

- Then, it acts as CA. Though, ATT_e does not know each server's private key, it knows these servers' identities. Therefore, it can use each server's identity to replace the original corresponding private key in the computation of $f_i(X)$ as shown in the following equation, Equation (4).

$$f_i(X) = \sum_{j=1}^{m}(U\_ID_i + E\_T_{ij})\frac{(X - U\_ID_i)}{(S\_ID_j - U\_ID_i)} \times \prod_{k=1,k\neq j}^{m}\frac{(X - S\_ID_k)}{(S\_ID_j - S\_ID_k)}$$

$$+ U\_R_i \prod_{y=1}^{m}\frac{(X - S\_ID_y)}{(U\_ID_i - S\_ID_y)} \pmod P$$

$$= a_m X^m + a_{m-1} X^{m-1} + \ldots + a_1 X + a_0 \pmod{P} \qquad \cdots\cdots\cdots\cdots \text{Equation (4)}$$

(2) In login stage, when ATT_e wants to login to server S_j, it performs the following steps:

- ATT_e gets a timestamp t from the system, then it generates a secret random number r_1', and computes C_1, C_2, and p as follows:

  $$C_1 = g^{r_1'} \pmod{P}$$

  $$C_2 = r_1' + r \cdot h(C_1, t) \pmod{P}$$

  $$p = (S\_ID_j)^{r_1'} \pmod{P}$$

- Then, ATT_e computes $f_i(1)$, $f_i(2)$, …, $f_i(m)$, and $f_i(p)$ and sends an authentication message $M = \{U\_ID_i, t, C_1, C_2, f_i(1), f_i(2), \ldots, f_i(m), f_i(p)\}$ to the server S_j.

(3)The server authentication stage

When receiving the authentication message from ATT_e, S_j records the current timesatamp in t_{now}. He then performs following verification steps to authenticate ATTe.

- checks ATT_e's inentity U_ID_i and whether or not $t_{now} - t > \Delta T$. If both hold, S_j computes

  $$p = (C_1)^{S\_SK_j} \pmod{P}.$$

- uses the received m + 1 points $\{(1, f_i(1)), (2, f_i(2)), \ldots, (m, f_i(m)), (p, f_i(p))\}$ from ATT_e to reconstruct the interpolating polynomial

  $$f_i(X) = a_m X^m + a_{m-1} X^{m-1} + \ldots + a_1 X + a_0 \pmod{P}.$$

- verifies whether or not $\dfrac{g^{C_2}}{(C_1) \cdot (U\_R_i)^{h(C_1, t)}} = 1$, if it holds, ATT_e is authentic.

Obviously, ATT_e can pretend as U_i successfully. Since the computation result of the verification is doomed to equal 1 as shown in the following equation, Equation (5).

$$\frac{g^{C_2}}{(C_1)\cdot(U\_R_i)^{h(C_1,t)}} = \frac{g^{r_1'+r*h(C_1,t)}}{g^{r_1'}\cdot g^{r*h(C_1,t)}}$$

$$= \frac{g^{r_1'+r*h(C_1,t)}}{g^{r_1'+r*h(C_1,t)}}$$

$$= 1 \pmod P \quad \cdots\cdots\cdots\cdots\cdots\cdots \text{Equation (5)}$$

The system setup stage

CA

1. chooses prime numbers P
   Chooses an one-way hash function $h(X,Y)$
   $g \in Z_P^*$
2. (For each server S_j, $1 \le j \le m$):
   chooses secret key S_SK_j
   computes S_j's identity $S\_ID_j = g^{S\_SK_j} \pmod P$

The user registration stage

ATT_e

1. chooses U_ID_i, U_PW_i
2. chooses random number r
   computes $U\_R_i = g^r \pmod P$
   computes $U\_S_i = r^{-U\_PW_i} \pmod P$
   uses S_j's identity to replace the original corresponding private key
   constructs

$$f_i(X) = \sum_{j=1}^{m}(U\_ID_i + E\_T_{ij})\frac{(X - U\_ID_i)}{(S\_ID_j - U\_ID_i)} \times \prod_{k=1,k\neq j}^{m}\frac{(X - S\_ID_k)}{(S\_ID_j - S\_ID_k)} + U\_R_i\prod_{y=1}^{m}\frac{(X - S\_SK_y)}{(U\_ID_i - S\_SK_y)} \pmod P$$

$$= a_m X^m + a_{m-1}X^{m-1} + \ldots + a_1 X + a_0 \pmod P$$

3.stores $U\_S_i, f_i(X)$ to ATT_e's storage device

Fig. 4. Attack on Tsaur et al.'s second protocol

The login stage

ATT$_e$ | S$_j$

1. keys in U_PW_i
2. gets a timestamp t

   generates a secret random number r_1'

   computes

   $C_1 = g^{r_1'} \pmod{P}$

   $C_2 = r_1' + r \cdot h(C_1, t) \pmod{P}$

   $p = (S\_ID_j)^{r_1'} \pmod{P}$
3. computes $f_i(1), f_i(2), \ldots, f_i(m)$, and $f_i(p)$
4. constructs an authentication message:

   $M = \{U\_ID_i, t, C_1, C_2, f_i(1), f_i(2), \ldots, f_i(m), f_i(p)\}$

$$\xrightarrow{M}$$

The server authentication stage

S$_j$

1. t_{now} is current timestamp, checks the identity of U_ID_i and whether or not $t_{now} - t > \Delta T$

   if U_ID_i is invalid or $t_{now} - t > \Delta T$, S$_j$ *rejects ;* otherwise, it continues

2.computes

   $p = (C_1)^{S\_SK_j} \pmod{P}$

2. uses points $\{(1, f_i(1)), (2, f_i(2)), \ldots, (m, f_i(m)), (p, f_i(p))\}$ to reconstruct

   $f_i(X) = a_m X^m + a_{m-1} X^{m-1} + \ldots + a_1 X + a_0 \pmod{P}$

3. checks whether $\dfrac{g^{C_2}}{(C_1) \cdot (U\_R_i)^{h(C_1, t)}} = 1$,

   if it holds, ATT$_e$ is verified. ATT$_e$ can pretend as U$_i$ successfully.

Fig. 4-continued Attack on Tsaur et al.'s second protocol

Chapter 4 Discussion

In this paper, we present the security analysis of Tsaur et al.'s two smart card based password authentication protocols in multi-server environments. Our results show that they are both vulnerable and suffer from the impersenation attack which we have described in this article.

References

[1] A. K. Awasthi and S. Lal "An enhanced remote user authentication scheme using smart cards, "*IEEE Trans. Consumer Electron.,* vol. 50, No. 2, pp. 583-586, May 2004.

[2] C.K. Chan, L.M. Cheng, "Cryptanalysis of a remote user authentication scheme using smarts cards, "*IEEE Transactions on Consumer Electronics,* vol. 46, no 4, pp. 992–993, 2000.

[3] C.C. Chang, J.S. Lee, "An efficient and secure multi-server password authentication scheme using smart cards", *Proceedings of International Conference on Cyberworlds*, No. 18-20, pp. 417-422, Nov 2004.

[4] C.C. Chang, T.C. Wu, "Remote password authentication scheme with smart cards, "*IEE Proceedings-Computers and Digital Techniques,* vol. 138, issue 3, pp.165–168, 1991.

[5] W. Diffie and M.E. Hellman, "New Directions in Cryptography, "*IEEE Transactions on Information Theory,* Vol. IT-22, No. 6, pp. 644-654, Nov. 1976.

[6] T. Hwang, Y. Chen, and C.S. Laih, "Non-interactive password authentications without password tables, "*IEEE Region 10 Conference on Computer and Communication Systems, IEEE Computer Society,* Vol. 1, pp.429–431, 1990.

[7] M.S. Hwang and L.H. Li, "A new remote user authentication scheme using smart cards, "*IEEE Transactions on Consumer Electronics,* vol.46, no.1, pp. 28-30, Feb. 2000.

[8] 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, Feb 2004.

[9] L. Lamport, "Password authentication with insecure communication, " *Communications of the ACM,* Vol. 24, No. 11, pp. 770-772, Nov. 1981.

[10] K. C. Leung, L. M. Cheng, A. S. Fong and C. K. Chan, "Cryptanalysis of a modified remote user authentication scheme using smart cards, "*IEEE Trans. Consumer Electron.,* vol. 49, No. 4, pp. 1243-1245, Nov. 2003.

[11] R.L. Rivest, A. Shamir, and L.M. Adleman, "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems, "*Communications of the ACM,* Vol. 21, No. 2, pp. 120-126, Feb. 1978.

[12] A. Shamir, "How to share a secret, "*Communications of the ACM,* Vol. 22, issus 11, pp. 612–613, Nov. 1979.

[13] J.J. Shen, C. W. Lin and M. S. Hwang, "A modified remote user authentication scheme using smart cards, "*IEEE Trans. Consumer Electron.,* vol. 49, No. 2, pp. 414-416, May 2003.

[14] W.J. Tsaur, C.C. Wu, W.B. Lee, “A smart card-based remote scheme for password authentication in multi-server Internet services, “*Computer Standards & Interfaces,* Vol. 27, No. 1, pp. 39-51, November 2004.

[15] W.J. Tsaur, C.C. Wu, W.B. Lee, “An enhanced user authentication scheme for multi-server Internet services, ”*Applied Mathematics and Computation*, Vol. 170, No. 1-1, pp. 258-266, November 2005.