

南 華 大 學

資訊管理學系

碩士論文

網路上的密碼確認及鑰匙建立方法之研究

The Research of Password Authentication and key

Establishment for Internet Protocols



研 究 生：楊明得

指 導 教 授：周志賢 博士

中 華 民 國 九 十 五 年 六 月

南 華 大 學
資 訊 管 理 學 系
碩 士 學 位 論 文

The Research of Password Authentication and key
Establishment for Internet Protocols


研究生：楊明得

經考試合格特此證明

口試委員：_____

林文忠
紀英華

指導教授：周志賢

系主任(所長)：

口試日期：中華民國 95 年 6 月 16 日

誌 謝

在這說長不長，說短不短的兩年研究生活中，首先我要感謝我的指導老師周志賢博士，因為有他的諄諄不誨的教導，才有今日的我。而他那嚴謹的指導作風，也造就了我有了一顆不怕困難的心及解決事情的能力，因此在面對未來更艱難的挑戰時，我十分的有信心能克服，因為我深信「沒有熬不過的苦，只有熬不過的人」。

而我也要感謝同研究室的夥伴、草莓、阿堂、阿輝及老涂，因為有了他們，我的研究生活才不至於平淡，且處處都充滿了樂趣，而我要感謝家卉，因為她的支持是我最大的動力。更要感謝阿聰，他總是幫我注意寫文章的格式。還要感謝所有曾經協助我的人，謝謝你們。最後更謝謝我的家人，謝謝你們永遠那麼支持著我。

楊明得 謹識

于 南華大學資管所

九十五年六月

網路上的密碼確認及鑰匙建立方法之研究

學生：楊明得

指導教授：周志賢

南 華 大 學 資 訊 管 理 學 系 碩 士 班

摘 要

在現今這個網路發達的世界裡，由於資訊大多是被公開的且容易被取得，因此為了保障一些重要的資訊，人們往往會採取許多的安全措施，其中又以密碼確認及協商鑰匙建立這兩種技術為最為常見。首先以密碼確認技術來說，在 2005 年時 Yang 和 Wang 兩人提出了一個結合晶片卡的密碼確認流程來確保遠端使用者的認證安全。不過他們的架構卻存在著安全上的弱點，也就是無法阻擋攻擊者的主動偽裝攻擊。而當確認完身分之後，就可以再運用鑰匙建立技術來協商出一把通訊用的鑰匙，以供之後彼此通訊時使用。而在 2003 年時 Boyd 和 Mao 兩人也發展了一個以橢圓曲線為主的鑰匙建立流程來保障安全的傳輸。然而，此方法也同樣存在著一些安全上的缺陷，就是容易遭受鑰匙遺失的假裝攻擊。

因此，在本篇論文我們將個別的分析 Yang 和 Wang 及 Boyd 和 Mao 等人的架構，指出其安全上的缺陷，並針對偽裝攻擊及鑰匙遺失的假裝攻擊來提出攻擊的演算法，最後更進一步的對遭受偽裝攻擊的 Yang 和 Wang 之密碼確認流程提出改進的方法，以增加其可信性。

關鍵字：密碼確認；協商鑰匙建立；晶片卡；偽裝攻擊；鑰匙遺失的
假裝攻擊

The Research of Password Authentication and key Establishment for Internet Protocols

Student : Ming-De Yang

Advisors : Dr. Jue-Sam Chou

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

ABSTRACT

Because most of information is public and acquired easily in this network fast-developing world, people usually take many safety measures to protect the important information. The password authentication and key establishment are the two most common technologies. At first, in the method of password authentication, Yang and Wang proposed password schemes with smart card to assure legal users can login in and access the resource of the server in 2005. But their schemes are vulnerable to an active attacker who will take forgery attack. After authenticating the identity, user then can make use of key establishment technology to establish the session key for follow communication. Boyd and Mao proposed key establishment protocols using elliptic curve pairings for Internet protocols in 2003. However, their protocols are also vulnerable to key-Compromise Impersonation attack

We will take cryptanalysis of Yang–Wang schemes and Boyd-Mao protocols and propose forgery attack algorithm for Yang–Wang schemes and key-Compromise Impersonation attack algorithm for Boyd-Mao protocols. Then we further improve the Yang–Wang schemes and make them be able to against forgery attack.

Keywords: password authentication, key establishment, smart card, forgery attack, key-Compromise Impersonation attack

Table of Contents

Chapter 1 Introduction	1
Chapter 2 Review Related Paper	4
2.1 Yang and Wang's Password Authentication Schemes.....	4
2.1.1 Timestamp-based password authentication scheme.....	4
2.1.2. Nonce-based password authentication scheme.....	6
2.2 Boyd-Mao Deniable Authenticated key Establishment Protocols.....	9
2.2.1 Bilinear Weil Pairing:.....	9
2.2.2 MAC based authenticator:	9
2.2.3 Boyd-Mao key establishment protocols.....	10
Chapter 3 Cryptanalysis on the Security.....	14
3.1 Yang and Wang's Password Authentication Schemes.....	14
3.1.1 Attack on the timestamp-based password authentication scheme	14
3.1.2 Attack on the nonce-based password authentication scheme	16
3.2 Boyd-Mao Deniable Authenticated key Establishment Protocols.....	19
3.2.1 Attack on the Key Establishment Using Diffie-Hellman Key Exchange.....	20
3.2.2 Attack on the Key Establishment based on Public Key Encryption Approach.....	21
Chapter 4 Improvement and Security Analysis	24
4.1 Our improvements on Yang and Wang's password authentication schemes ...	24
4.1.1 Improvement on the timestamp-based password authentication scheme.....	24
4.1.2 Improvement on the nonce-based password authentication scheme .	26
4.2 Security analysis	27
4.2.1 Forgery attack	28
4.2.2 Password-guessing attack	28
4.2.3 Replay attack.....	29
4.2.4 Smart card loss	29
Chapter 5 Conclusion.....	31
References.....	32

Table of Figures

Figure 1: Yang and Wang’s timestamp-based password authentication scheme	6
Figure 2: Yang and Wang’s nonce-based password authentication scheme	8
Figure 3: MAC based authenticator	10
Figure 4: Boyd-Mao key establishment using Diffie-Hellman key exchange.....	12
Figure 5: Boyd-Mao key establishment from public key encryption approach.....	13
Figure 6: Our attack on timestamp-based password authentication scheme.....	16
Figure 7: Our attack on nonce-based password authentication scheme	19
Figure 8: KCI attack on the Key Establishment Using Diffie-Hellman key exchange	21
Figure 9: KCI attack on the Key Establishment based on Public Key Encryption	23
Figure 10: Our improvement on Yang and Wang’s timestamp-based password authentication scheme	25
Figure 11: Our improvement on Yang and Wang’s nonce-based password authentication scheme	27

Chapter 1 Introduction

Because most of information is public and acquired easily in this network fast-developing world, people usually take many safety measures to protect the important information. The password authentication and key establishment are the two most common technologies. Password authentication is that a legal user can be ensured logging in to a remote server and accessing the authorized data. Recently, several password authentication schemes using smart card which has storage and computation abilities have been proposed [2, 10, 13, 24, 25 26, 27].

In 1999, Yang and Shieh [25] proposed both of a timestamp-based and a nonce-based password authentication schemes with smart cards first. They claimed that their scheme not only needn't to hold a verified table of passwords but also allow the users to select or change passwords freely. However in 2002, Chan and Cheng [4] found that their schemes were vulnerable to both of the given-ciphertext and forgery attacks. But in 2003, Sun and Yeh [10] indicated that Chan and Cheng's attack was unreasonable since the client's ID forged did not exist in the ID table, meanwhile they also showed that Yang and Shieh's schemes were subject to the forgery attack. Moreover, also in the same year Chen and Zhong [15] pointed out that the fundamental computation assumption in the Yang and Shieh's schemes were incorrect. After that, Jian and Pan [19] made a further analysis on Yang and Shieh's protocols in 2004 and deduce that their protocols are not secure. To overcome these problems, in 2005, Yang and Wang [3] proposed password schemes with smart card to resist the existing attacks. But we find that their schemes are

vulnerable to an active attacker who will take forgery attack.

After authenticating the identity, user then can make use of key establishment technology to establish the session key for follow secure communication [16]. Key establishment denotes a protocol whereby two parties jointly establish a secret key by communicating over a public channel. One of the basic secure communication technologies is the key establishment protocol that is known as Internet Key Exchange (IKE). It is the standard of Internet protocol Security (IPSec) proposed by the IETF in 1998[6, 14, 17, 18, 21].But, people have many criticisms for this protocol, especially for its complexity [8, 17].

In order to overcome such a problem, the elliptic curve cryptography that can reduce the computations and maintain the same security level becomes a better choice [1, 8, 9, 11, 12, 30]. Therefore in recent years, several cryptography schemes [5, 7, 8, 9, 11, 12, 20, 22, 28, 29, 30] are designed based on the elliptic curve. One of these schemes is the deniable authenticated key establishment for Internet protocols proposed by Boyd and Mao [8] in 2003. For the use of the elliptic curve cryptography, their schemes not only solve the complexity of computation but also become more efficient than others. However, we will also point out that Boyd-Mao deniable authenticated key establishment for Internet protocols can't resist against the key-Compromise Impersonation (KCI) attack defined by Wilson and Menezes [23]. The attack means that if A's long-term secret key is compromised and known by an adversary, the adversary can pretend others to communicate with A.

We also further improve the Yang-Wang schemes after taking cryptanalysis of them and make them become more security. The

structure of this thesis is organized as follows. Chapter 2 is review of Yang–Wang schemes [3] and Boyd-Mao protocols [8]. Chapter 3 is cryptanalysis of Yang–Wang schemes and Boyd-Mao protocols. Chapter 4 is our improvement and security analysis of Yang–Wang schemes. Chapter 5 is the conclusion.

Chapter 2 Review Related Paper

2.1 Yang and Wang's Password Authentication Schemes

In this section, we will introduce both of Yang and Wang's timestamp-based and nonce-based password authentication schemes [3]. In their schemes, there exists a key information center (KIC) whose responsibilities are to generate key information, issue smart cards to new users, and change passwords for the users if needed. And each scheme can be divided into three phases, registration phase, login phase and authentication phase. We will describe both of them as follows

2.1.1 Timestamp-based password authentication scheme

(i). Registration phase

After a new user U_n gives his identifier ID_n and password PW_n to the KIC through a secret channel, KIC executes the following steps.

Step1: Selects two large primes, p and q , computes $m = p * q$, where "*" denotes the multiplication operation and then selects a public key e and finds its corresponding secret key d satisfying $e * d \equiv 1 \pmod{\phi(m)}$.

Step2: Finds an integer g which is a primitive root in both $GF(p)$ and $GF(q)$.

Step3: Produces a smart card's identifier CID_n for user U_n , computes two

parameters, $S_n = ID_n^{CID_n \cdot d} \bmod m$ and $h_n = g^{PW_n \cdot d} \bmod m$, then issues the smart card, which includes $(m, e, g, ID_n, CID_n, S_n, h_n)$, to the user.

(ii). Login phase

When U_n wants to login to the remote server, he inserts his smart card into the input device and keys in his ID_n and PW_n . The smart card then executes the following steps:

Step1 : Produces a random number R_n and computes two parameters, $X_n = g^{PW_n \cdot R_n} \bmod m$ and $Y_n = S_n \cdot h_n^{R_n \cdot T} \bmod m$, where T is the current time of the input device.

Step2 : Sends the login message M which consists of $(ID_n, CID_n, X_n, Y_n, m, e, g, T)$ to the remote server

(iii). Authentication phase

After receiving the login message M from U_n , the remote server records the current time T' and executes the following steps:

Step1 : Checks to see whether ID_n and CID_n are right. If they are wrong, the login request will be rejected.

Step2 : Checks to see whether $(T' - T)$ is within a specified time interval ΔT . If it is, the request is legal; otherwise, the login request will be rejected.

Step3: Checks to see whether the equation $Y_n^e \equiv ID_n^{CID_n} \cdot X_n^T \pmod m$ holds. If it holds, the remote server accepts the login request. The figure of Yang and Wang's timestamp-based password authentication scheme is shown in Figure 1.

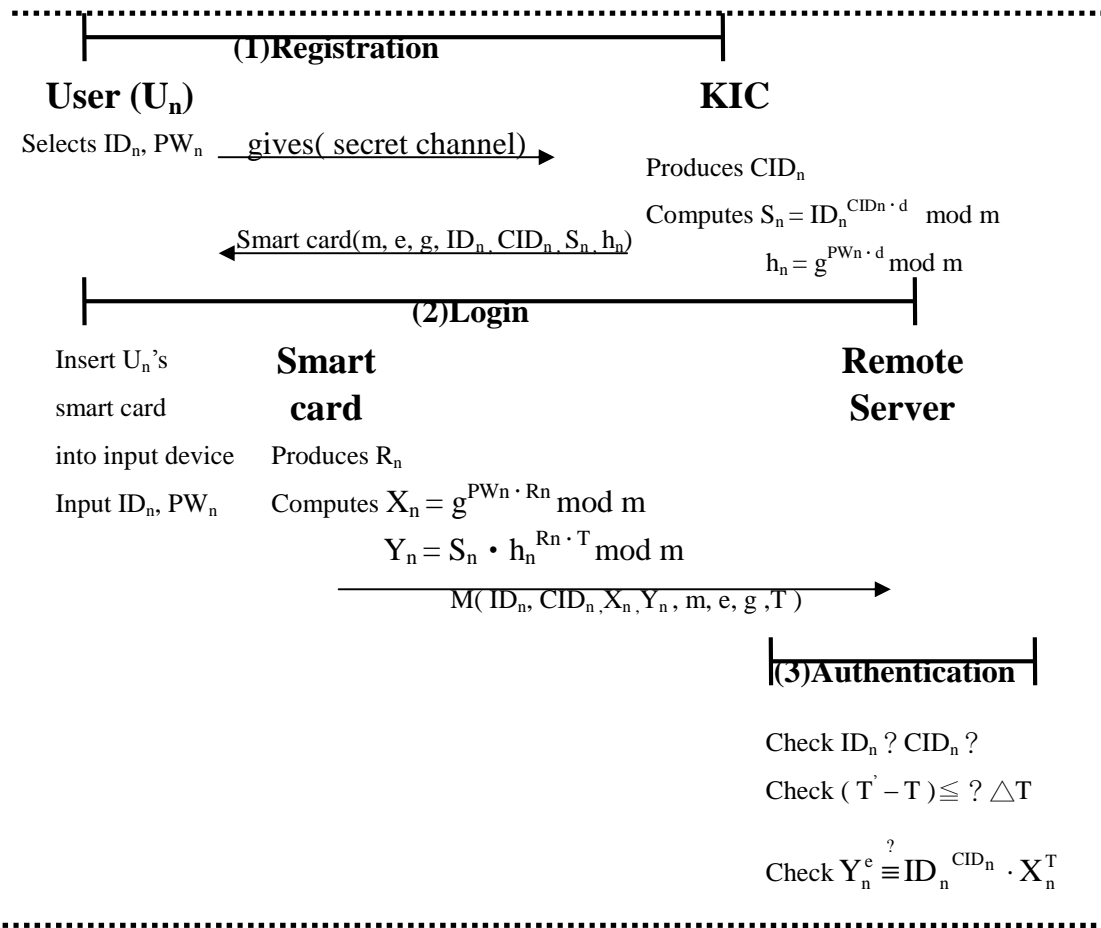


Figure 1: Yang and Wang's timestamp-based password authentication scheme

2.1.2. Nonce-based password authentication scheme

(i). Registration phase

After user U_n gives his identifier ID_n and password PW_n to the KIC

through a secret channel, the KIC executes the following steps:

Step1: Produces two large primes, p and q , computes $m = p * q$, where

"*" denotes the multiplication operation and then selects a public key e and finds its corresponding secret key d satisfying $e * d \equiv 1 \pmod{\phi(m)}$.

Step2: Finds an integer g which is a primitive root in both $GF(p)$ and $GF(q)$.

Step3: Produces a smart card's identifier CID_n for U_n , computes two parameters, $S_n = ID_n^{CID_n \cdot d} \pmod{m}$ and $h_n = g^{PW_n \cdot d} \pmod{m}$, then issues the smart card, which includes $(m, e, g, ID_n, CID_n, S_n, h_n)$, to the user.

(ii). Login phase

When user U_n wants to login to the remote server, he inserts his smart card into the input device and keys in his ID_n and PW_n . Then the smart card and the remote server together execute the following steps.

Step1: The smart card delivers the login message M_1 which consists of ID_n and CID_n to the remote server.

Step2: After receiving the login message M_1 , the remote server checks to see whether ID_n and CID_n are right. If they are right, the remote server selects a random number R_s , computes a nonce $N = h(R_s)$ and then delivers it to the smart card, where $h(.)$ denotes a one-way hash function.

Step3: The smart card produces a random number R_n and computes two

parameters, $X_n = g^{\text{PW}_n \cdot R_n} \text{ mod } m$ and $Y_n = S_n \cdot h_n^{R_n \cdot N} \text{ mod } m$.

Then the smart card delivers the message M_2 that consists of (X_n, Y_n, m, e, g) to the remote sever.

(iii). Authentication phase

After receiving message M_2 , the remote sever computes to see whether the equation $Y_n^e \equiv \text{ID}_n^{\text{CID}_n} \cdot X_n^N \text{ mod } m$ holds. If it so, the remote server accepts the login request; otherwise, it rejects. The figure of Yang and Wang’s nonce-based password authentication scheme is shown in Figure 2.

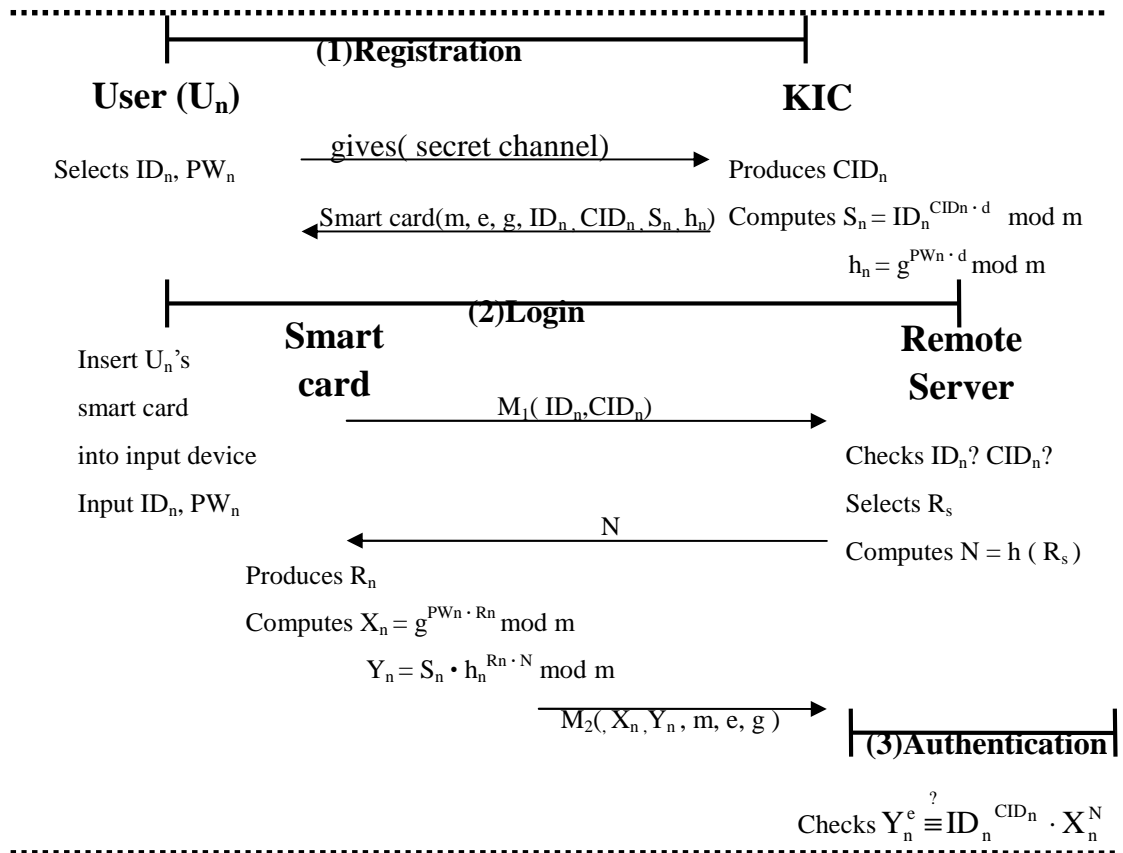


Figure 2: Yang and Wang’s nonce-based password authentication scheme

2.2 Boyd-Mao Deniable Authenticated key Establishment Protocols

In this section, we review Boyd-Mao deniable authenticated key establishment protocols. First, we will introduce pairings on elliptic curves. Next we will introduce MAC based authenticator. At last, we present the Boyd-Mao key establishment protocols.

2.2.1 Bilinear Weil Pairing:

Let G_1 be an additive group and G_2 be a multiplicative group and each of them have the same order. Then we assume that there exists an efficient computable bilinear map e , which is defined as $e: G_1 \times G_1 \rightarrow G_2$ and satisfies the following conditions:

1. Bilinear: For any $a, b \in \mathbb{Z}$ and $P, Q, R \in G_1$, we have $e(aP, bQ) = e(P, Q)^{ab}$ and $e(P, Q + R) = e(P, Q) \cdot e(P, R)$ and $e(P + Q, R) = e(P, R) \cdot e(Q, R)$.
2. Non-degenerate: For any $P, Q \in G_1$, we have $e(P, Q) \neq 1$.
3. Computability: For any $P, Q \in G_1$, there is an efficient algorithm to compute $e(P, Q) \in G_2$.

2.2.2 MAC based authenticator:

To construct the authenticator, user B first chooses a random number N_B and sends it to user A. When A receives N_B , he chooses an intended

message m and sends it together with $\text{MAC}_{F_{AB}}(B, N_B, m)$ to B, where B is user B's ID that is public and the MAC key F_{AB} can be non-interactively computed as the session key shared by both A and B. That is, A can compute F_{AB} as $e(sQ_A, Q_B)$ and B can compute $F_{BA}(=F_{AB})$ as $e(Q_A, sQ_B)$. The figure of MAC based authenticator is shown in Figure 3.

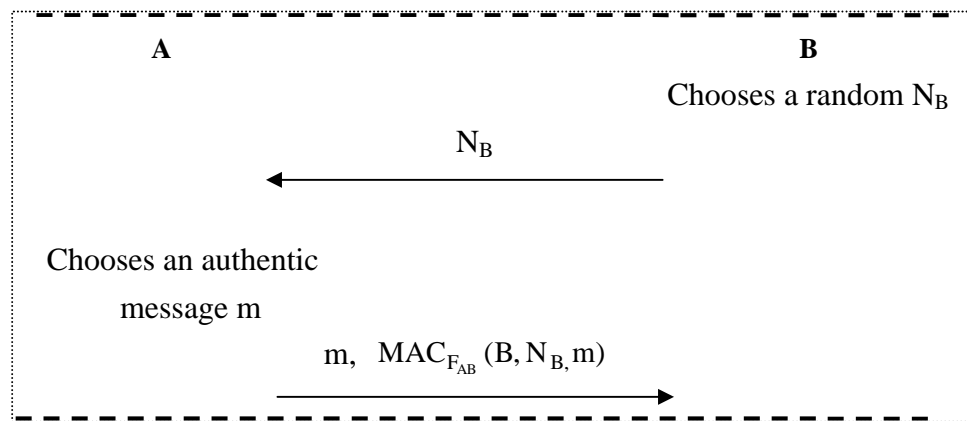


Figure 3: MAC based authenticator

2.2.3 Boyd-Mao key establishment protocols

The Boyd-Mao key establishment protocols can be divided into two portions, one is the key establishment protocol established using Diffie-Hellman key exchange, and the other key is established from public key encryption approach. Each of them can be stated as follows:

(i). Key Establishment Using Diffie-Hellman Key Exchange

In this scheme, a key exchange between users A and B can be accomplished as follow:

Users A and B each chooses a random number R_a and R_b respectively, then they compute g^{R_a} and g^{R_b} individually, where R_a, R_b

belongs to Z_q and g is a primitive root. In the protocol, users A's and B's IDs are ID_A and ID_B respectively, and F_{AB} denotes the non-interactively computed MAC key shared by both A and B derived from the bilinear pairing computation. That is, A can compute F_{AB} as $e(sQ_A, Q_B)$ and B can compute $F_{BA} (= F_{AB})$ as $e(Q_A, sQ_B)$. After that, A and B can begin to exchange information. The steps are as follows:

Step1. User A sends $t_A = g^{Ra}$ to user B. After accepting t_A , B will send $t_B = g^{Rb}$ and $MAC_{F_{AB}}(ID_B, t_A, t_B)$ to user A.

Step2. When user A receives t_B and $MAC_{F_{AB}}(ID_B, t_A, t_B)$, he can verify whether the MAC is authentic. If it is authentic, he will send $MAC_{F_{AB}}(ID_A, t_B, t_A)$ to user B. Then A can compute the final session key $Z_{AB} = t_B^{Ra}$ shared with B.

Step3. After accepting the $MAC_{F_{AB}}(ID_A, t_B, t_A)$, B will verify whether the MAC is authentic. If it is authentic, B then computes the session key $Z_{BA} (= Z_{AB} = t_A^{Rb})$. The figure of Boyd-Mao key establishment using Diffie-Hellman key exchange shown in Figure 4

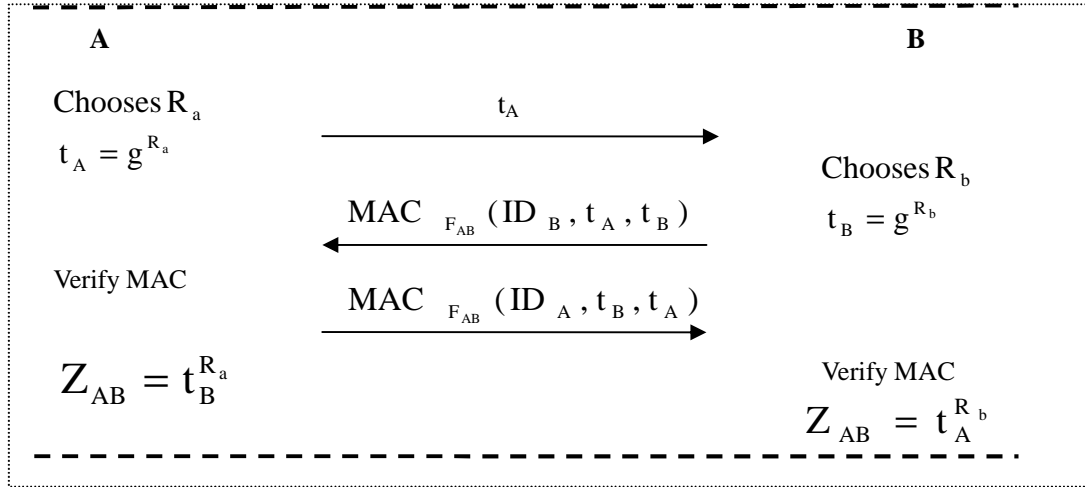


Figure 4: Boyd-Mao key establishment using Diffie-Hellman key exchange

(ii). Key Established from Public Key Encryption Approach

In this scheme, users A and B each chooses a random number N_A and N_B respectively, where $N_A, N_B \in [1 \dots t]$. F_{AB} denotes the same value defined in section 2.2.3.(i). Then, A and B can begin to exchange information. The steps are as follows:

Step1. User B sends N_B to user A. After receiving N_B , A chooses a session key K and encrypts it using B's public key denoted as $E_B(K)$. Then A sends $E_B(K)$, ID_A , N_A , and $MAC_{F_{AB}}(ID_B, N_B, E_B(K))$ to B.

Step2. When B receives $E_B(K)$, ID_A , N_a , and $MAC_{F_{AB}}(ID_B, N_B, E_B(K))$, he decrypts $E_B(K)$ with his private key to get K and using the MAC key F_{AB} to verify whether the MAC holds. If it holds, B can confirm he is communicating with A and sends $MAC_K(ID_A, ID_B, N_A, N_B)$ to user A.

Step3. After receiving $MAC_K(ID_A, ID_B, N_A, N_B)$, user A verifies whether the MAC holds. If it holds, the MAC is authentic and A can confirm he is communicating with the intended person B. Therefore, user A and B can begin to communicate with each other. The figure of Boyd-Mao key establishment from public key encryption approach is shown in Figure 5.

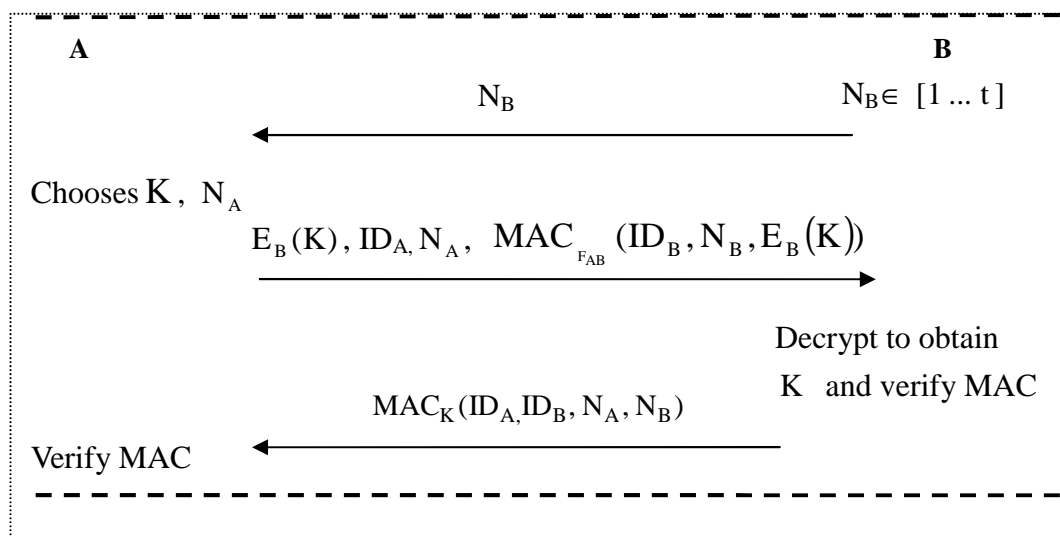


Figure 5: Boyd-Mao key establishment from public key encryption approach

Chapter 3 Cryptanalysis on the Security

3.1 Yang and Wang's Password Authentication Schemes

In this section, we will demonstrate that both of Yang and Wang's password authentication schemes suffer from the forgery attack. Since we can obtain some of the user's information from the login message and forge the required information to satisfy the verifying process in the authentication phase. We show both of our attacks as follows.

3.1.1 Attack on the timestamp-based password authentication scheme

By monitoring on the communication line, an attacker U_a can launch an attack by performing the following steps.

Step1: U_a intercepts the legal user's login message $M=(ID_n, CID_n, X_n, Y_n, m, e, g, T)$.

Step2: Since the verifying equation $Y_n^e \equiv ID_n^{CID_n} \cdot X_n^T \pmod m$ in the authentication phase can be transformed into $Y_n^e \cdot (X_n^T)^{-1} \equiv ID_n^{CID_n} \pmod m$, U_a can know the fixed value $ID_n^{CID_n}$ of user U_n , where the inverse of X_n^T , $(X_n^T)^{-1}$, can be calculated using the extended Euclid algorithm if it is relatively prime to m . Here on, we use the denominator of an element b to represent its multiplicative inverse, b^{-1} , modulus m .

Step3: Then U_a can base on message M got from step1 to substitute the

values of X_n , Y_n and T by the forged values X'_n , Y'_n and T' respectively, which can be computed as follows:

$$\begin{aligned}
ID_n^{CID_n} &= \frac{(Y_n)^e}{(X_n)^T} \bmod m \\
&= \frac{(K^T Y_n)^e}{(K^e X_n)^T} \bmod m \\
&= \frac{(K^T Y_n)^e \cdot (K^{eT} X_n^T)^e}{(K^e X_n)^T \cdot (K^e X_n)^{eT}} \bmod m \\
&= \frac{(K^T Y_n \cdot K^{eT} X_n^T)^e}{(K^e X_n)^{T+eT}} \bmod m \\
&= \frac{(K^{T(1+e)} X_n^T Y_n)^e}{(K^e X_n)^{T+eT}} \bmod m \\
&= \frac{(Y'_n)^e}{(X'_n)^{T'}} \bmod m
\end{aligned}$$

That is, U_a can first choose any random K such that $X'_n = K^e X_n$ has an inverse modulus m , and then let $Y'_n = K^{T(1+e)} X_n^T Y_n$ and $T' = T + eT$ to satisfy the verifying equation. After that, U_a can deliver the forged login message $(ID_n, CID_n, X'_n, Y'_n, m, e, g, T')$ to the remote server. Since the equation $(Y'_n)^e \equiv ID_n^{CID_n} \cdot (X'_n)^{T'} \bmod m$ holds as well in the authentication phase, the attacker U_a can thus impersonate the legal user U_n and subsequently cheat the remote sever successfully. Therefore, we have a successful attack in this timestamp-based password authentication

scheme. The figure of our attack on timestamp-based password authentication scheme is shown in Figure 6.

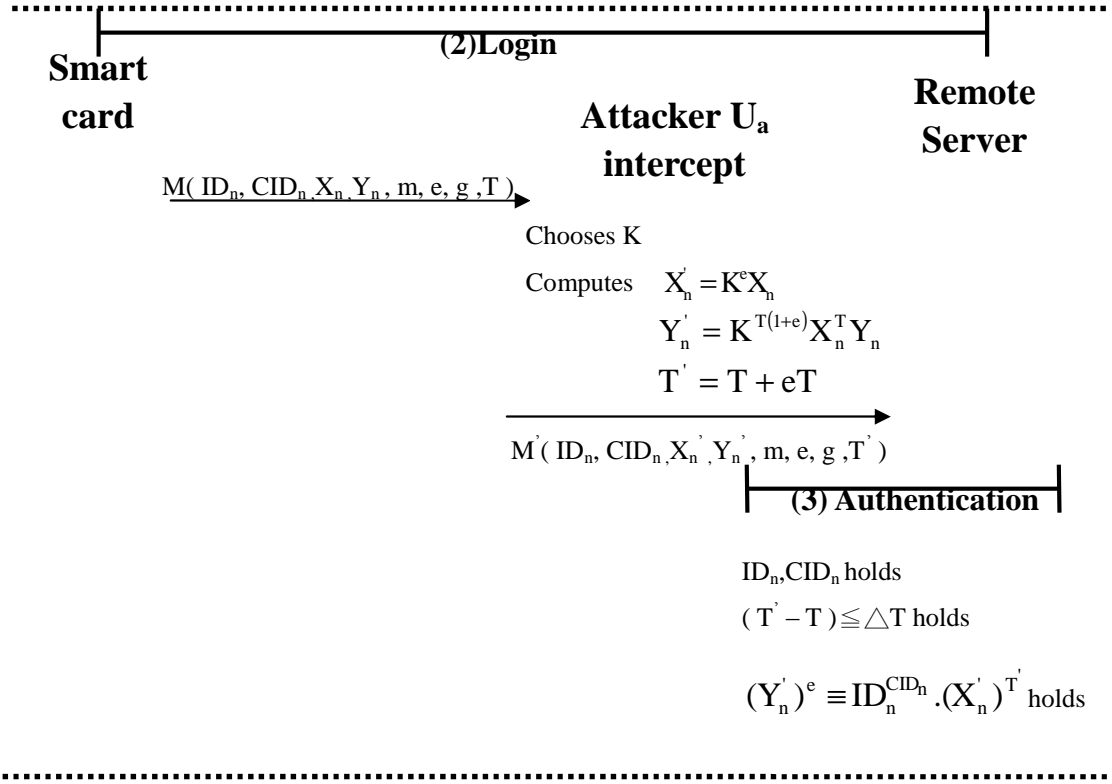


Figure 6: Our attack on timestamp-based password authentication scheme

3.1.2 Attack on the nonce-based password authentication scheme

In this attack, U_a can eavesdrop on the communication line to intercept the messages transformed between the user and the remote server for an enough period of time. Suppose that there are several login users recorded in the hacker's (U_a) database and each occurs several times during this period. More precisely, the hacker's database now contains the items for each login user (U_n) in the form $[(ID_n, CID_n), N_{nj}, (X_{nj}, Y_{nj}, m, e, g)]$, where the subindex nj denotes U_n 's j th login, ($ID_n,$

CID_n) is the message M_1 sent by the smart card to the server, N_{nj} is the server's responding random number, $(X_{nj}, Y_{nj}, m, e, g)$ is the message M_2 sent by the smart card to the server. After constructing this hacker database, U_a can then launch an attack by first finding an user U_i who has the most logged items in the database and then perform the following steps:

Step1: U_a sends U_i 's ID_i and CID_i to the remote server. Assume that the remote server responds with a random number N' back, then U_a might be able to find one of all U_i 's recorded items in which N_{ij} is smaller than N' and satisfies both $X_{nj}^{N_{nj}}$ modulus m has an inverse and $N' - N_{ij} = \Delta N = B \cdot e \cdot N_{ij}$, where $B \in Z_m$ and smaller than ΔN . If U_a can't find an user with such a N_{ij} , he would find the second, the third, etc, according to the user's occurrence frequency in the database until he can find such an user. In fact, U_a can decide whether $X_{nj}^{N_{nj}}$ modulus m has an inverse in his off-time precomputation stage (Here, we assume the found user is U_n .)

Step2: U_a then extracts the found user U_n 's selected item, $(ID_n, CID_n, N_{nj}, X_{nj}, Y_{nj}, m, e, g)$, in his hacker's database. For abbreviation, we denote U_n 's N_{nj}, X_{nj}, Y_{nj} as N, X_n, Y_n respectively in the following. Since the equation $Y_n^e \equiv ID_n^{CID_n} \cdot X_n^N \pmod{m}$ in the authentication phase can be transformed

into $\frac{Y_n^e}{X_n^N} \equiv \text{ID}_n^{\text{CID}_n} \pmod{m}$. So U_a can know the fixed value of the

$\text{ID}_n^{\text{CID}_n}$ in this congruence according to this selected item.

Step3: U_a substitutes the values of X_n, Y_n, N by the forged values X'_n, Y'_n , and the server's responding value N' respectively, which can be computed as follows:

$$\begin{aligned}
\text{ID}_n^{\text{CID}_n} &= \frac{Y_n^e}{X_n^N} \pmod{m} \\
&= \frac{(K^N Y_n)^e}{(K^e X_n)^N} \pmod{m} \\
&= \frac{(K^N Y_n)^e \cdot (K^e X_n)^{\text{Be}N}}{(K^e X_n)^N \cdot (K^e X_n)^{\text{Be}N}} \pmod{m} \\
&= \frac{(K^N Y_n)^e \cdot [(K^e X_n)^{\text{BN}}]^e}{(K^e X_n)^N \cdot (K^e X_n)^{\text{Be}N}} \pmod{m} \\
&= \frac{[K^N Y_n \cdot (K^e X_n)^{\text{BN}}]^e}{(K^e X_n)^{(1+\text{Be})N}} \pmod{m} \\
&= \frac{(Y'_n)^e}{(X'_n)^{N'}} \pmod{m}
\end{aligned}$$

That is, U_a can randomly choose any $K \in \mathbb{Z}_m$ such that $X'_n = K^e X_n$ has an inverse modulus m and then let $Y'_n = K^N Y_n \cdot (K^e X_n)^{\text{BN}}$ and $N' = (1 + \text{Be})N$ to satisfy the verifying equation. After that, U_a can deliver this forged login message $(X'_n, Y'_n, m,$

e, g) to the remote server. Since the equation $(Y'_n)^e \equiv ID_n^{CID_n} \cdot (X'_n)^{N'}$ mod m holds in the authentication phase as well, the attacker U_a can thus impersonate the legal user U_n and cheat the remote sever successfully. Hence, we also have a successful attack in the nonce-based password authentication scheme. The figure of our attack on nonce-based password authentication scheme is shown in Figure 7.

..

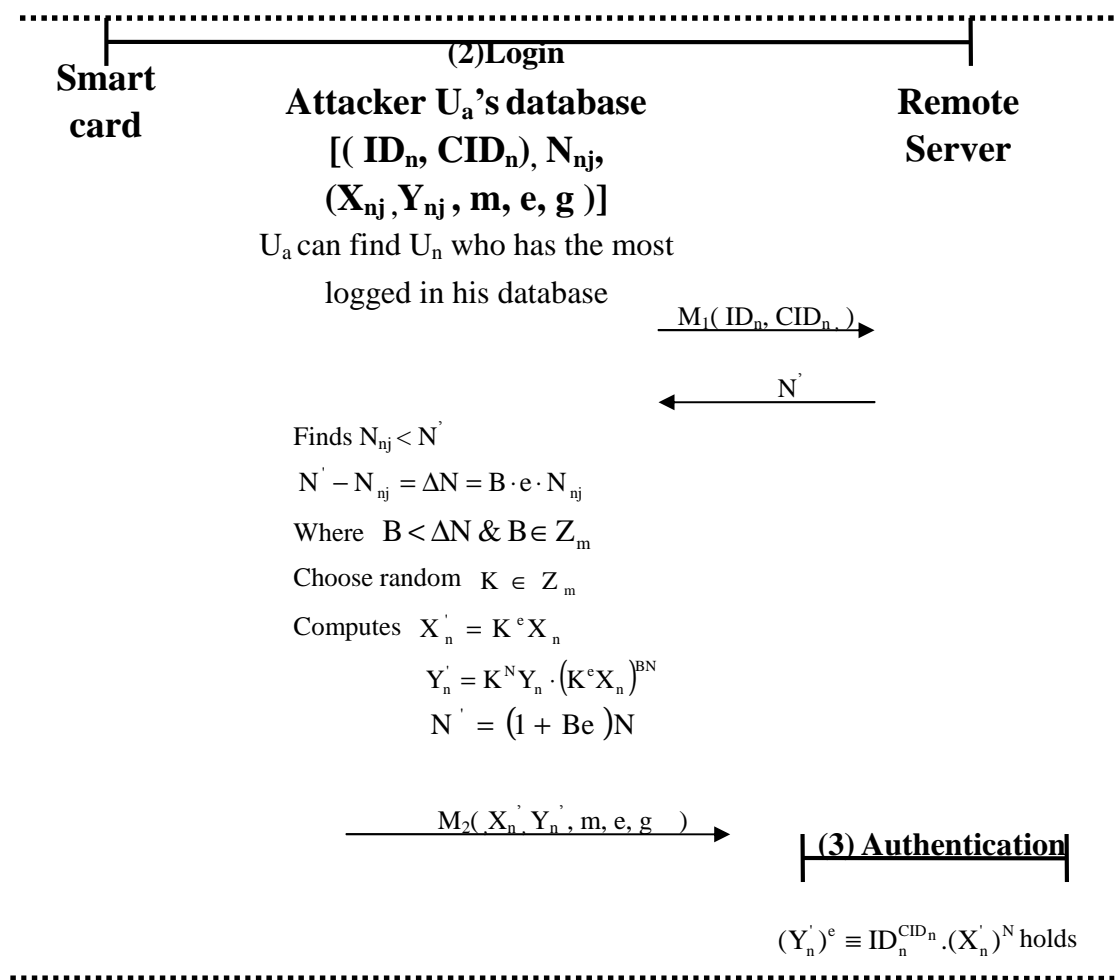


Figure 7: Our attack on nonce-based password authentication scheme

3.2 Boyd-Mao Deniable Authenticated key Establishment Protocols

In this section, we use the four security attributes defined by Wilson and Menezes [23] to analyze Boyd-Mao key establishment protocols. After that, we can find that Boyd-Mao key establishment protocols can't resist against the KCI attack. An adversary can pretend others to communicate with A when he obtains A's long-term secret key. Now, we show our KCI attacks on the Boyd-Mao key establishment protocols as follows:

3.2.1 Attack on the Key Establishment Using Diffie-Hellman Key Exchange

We assume an adversary X who knows user A's long-term secret key sQ_A and wants to launch the KCI attack to pretend user B to communicate with A. He can act as follows:

Step1. When X intercepts t_A sent from A intended to B, X can compute the MAC key F_{AB} in the same manner specified in Section 2.2.3(i) and choose a random number R_b' to compute $t_B' = g^{R_b'}$. Then he can send t_B' and $MAC_{F_{AB}}(ID_B, t_A, t_B')$ to user A.

Step2. After receiving t_B' and $MAC_{F_{AB}}(ID_B, t_A, t_B')$ from X, user A can verify it as authentic for he also has the MAC key F_{AB} . And because he knows t_B' , he can compute the session key $Z_{AB} = ((t_B')^{R_a})$ by Diffie-Hellman key exchange protocol, where $R_a \in_R Z_q$ is selected by A. After that, user A sends $MAC_{F_{AB}}(ID_A, t_B', t_A)$ back to X.

Step3. When X receives $MAC_{F_{AB}}(ID_A, t_B', t_A)$, he can also verify it successfully, because t_B' is computed by himself. So he can take t_A and his secret random R_b' to compute the session key $Z_{AB} = t_A^{R_b'}$. Accordingly, user A and X have the same session key and thus can communicate with each other. Because adversary X can send his message using B's ID, A will believe that he is communicating with B. So, adversary X can pretend to be user B to communicate with A successfully. Therefore, we have a successful KCI attack. The figure of KCI attack on this scheme is shown in Figure 8.

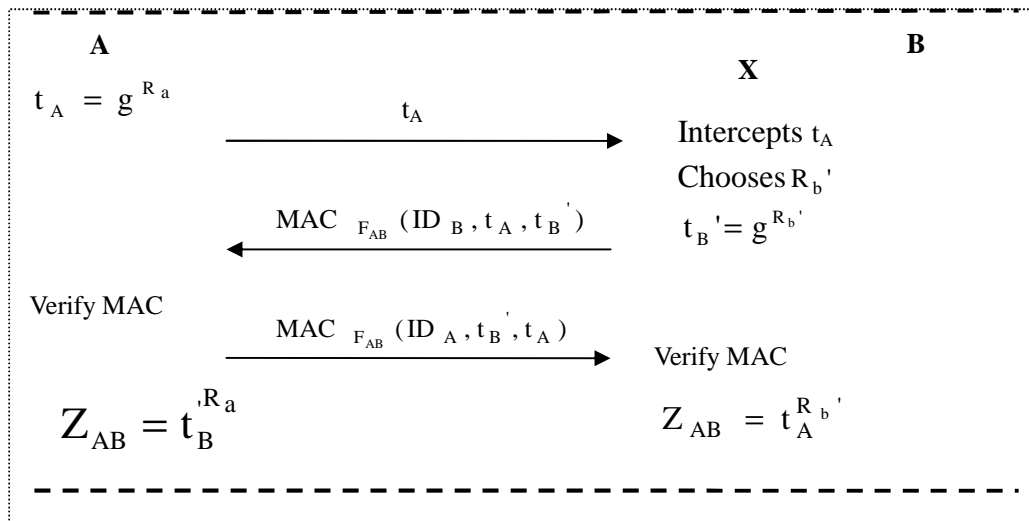


Figure 8: KCI attack on the Key Establishment Using Diffie-Hellman key exchange

3.2.2 Attack on the Key Establishment based on Public Key Encryption Approach

Here, we assume that an adversary X knows user B's long-term

secret key sQ_B . Under this assumption, when he wants to launch a KCI attack, he can compute the MAC key F_{AB} to pretend user A to communicate with B. We delineate it as follows:

Step1. After intercepting N_B sent by B, X will choose a random key

K' as the shared session key and encrypts it using B's public key denoted as $E_B(K')$. He also chooses a random number $N_{A'}$.

After that, he sends $ID_A, N_{A'}, E_B(K')$ and the computed $MAC_{F_{AB}}(ID_B, N_B, E_B(K'))$ to B.

Step2. After receiving the $E_B(K'), ID_A, N_{A'}$, and

$MAC_{F_{AB}}(ID_B, N_B, E_B(K'))$, B will decrypt $E_B(K')$ to get K' using his private key and verify to see if $MAC_{F_{AB}}(ID_B, N_B, E_B(K'))$ is authentic. Obviously, B will

verify it successfully for he also has the same MAC key F_{AB} as X does. After that B will send the authenticator encrypted with

the session key K' selected by X and send

$MAC_{K'}(ID_A, ID_B, N_{A'}, N_B)$ to user X. Then X can also verify

it successfully for K' is selected by himself.

Step3. Then users B and X will have the same session key K' , and thus

can communicate with each other. Because X sends his information using A's ID, B will believe that he is communicating with A. So X can pretend user A to communicate with B successfully. Therefore, we also have a

successful KCI attack. The figure of KCI attack on this scheme is shown in Figure 9.

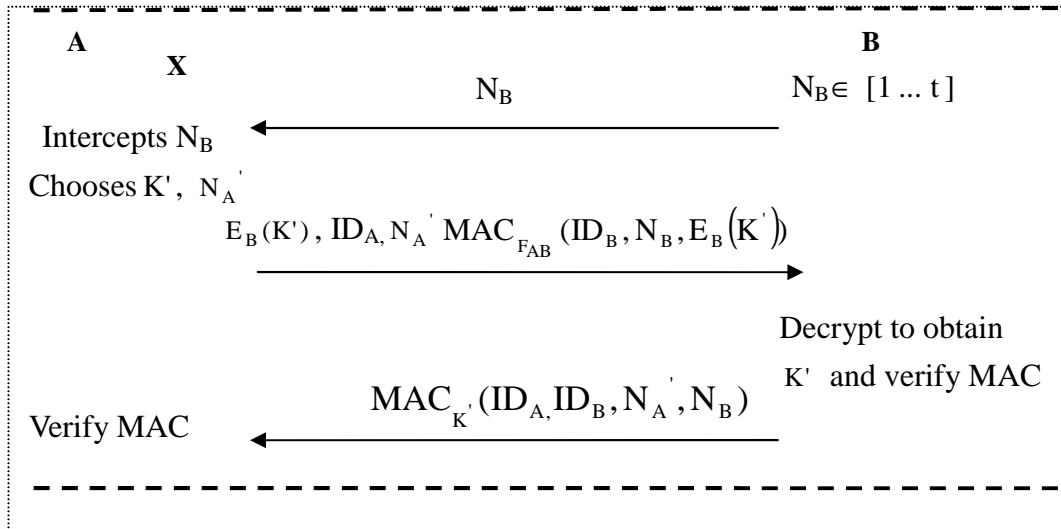


Figure 9: KCI attack on the Key Establishment based on Public Key Encryption

Chapter 4 Improvement and Security Analysis

4.1 Our improvements on Yang and Wang's password authentication schemes

4.1.1 Improvement on the timestamp-based password authentication scheme

(i). Registration phase

This phase is the same as the registration phase in the time-based password authentication scheme described in Section 2.1.1.(i).

(ii). Login phase

The user first inserts his smart card into the device and keys in his ID_n and PW_n when he wants to login in to the remote server. Then the smart card executes the following steps.

Step1 : Produces a random number R_n and computes the two formulas

that are $X_n = g^{PW_n \cdot R_n} \text{ mod } m$ and $Y_n = (S_n \cdot h_n^T)^{R_n} \text{ mod } m$,

where T is the current time of the input device.

Step2 : Sends the login message M which consists of $(ID_n, g^{CID_n},$

$CID_n * R_n, X_n, Y_n, m, e, g, T)$ to the remote server, where "*" denotes a multiplication operation.

(iii). Authentication phase

After receiving the login message M from the input device, the remote server records the current time T' and executes the following steps.

Step1 : Looks over whether the ID_n and CID_n are right or not and if they are wrong , then the login request is refused.

Step2 : Looks over whether $(T' - T)$ is within a specified time interval ΔT . If it is, then the request is legal, else the login request is refused.

Step3 : Looks over whether the equation $Y_n^e \equiv ID_n^{CID_n * R_n} \cdot X_n^T \pmod m$ holds or not. If it holds, the remote server accepts the login request. The figure of our improvement on Yang and Wang's timestamp-based password authentication scheme is shown in Figure 10.

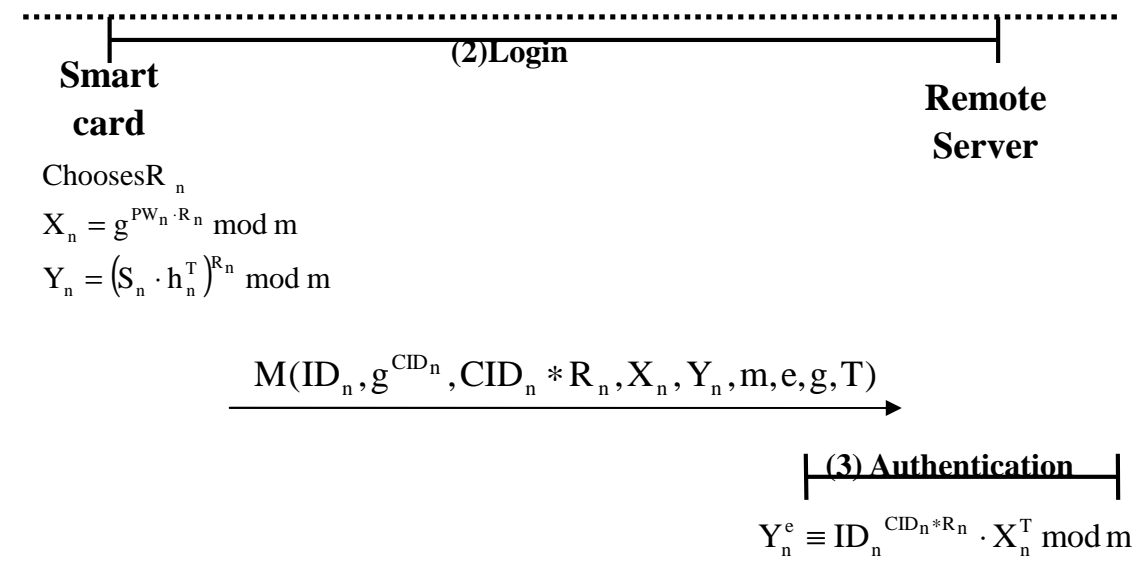


Figure 10: Our improvement on Yang and Wang's timestamp-based password authentication scheme

4.1.2 Improvement on the nonce-based password authentication scheme

(i). Registration phase

This phase is the same as the registration phase in the nonce-based password authentication scheme described in Section 2.2.1.(i).

(ii). Login phase

The user first inserts his smart card into the input device and keys in his ID_n and PW_n when he wants to login in to the remote server. Then the smart card executes the following steps.

Step1: The smart card delivers a login message M_1 that consists of ID_n and g^{CID_n} to the remote server.

Step2: The remote server looks over whether the ID_n and CID_n are right or not after it receives the login message M_1 . If they are right, the remote server computes a random number R_m and computes a nonce $N = h(R_m)$, then delivers it back to the smart card. The $h(.)$ is a one way hash function.

Step3: The smart card produces a random number R_n and computes two formulas that are $X_n = g^{PW_n \cdot R_n} \text{ mod } n$ and $Y_n = (S_n \cdot h_n^N)^{R_n} \text{ mod } m$. Then the smart card delivers the message M_2 that consists of $(X_n, Y_n, CID_n * R_n, m, e, g)$ to the remote server.

(iii). Authentication phase

The remote server computes whether the equation $Y_n^e \equiv ID_n^{CID_n \cdot R_n} \cdot X_n^N \pmod{m}$ holds after it receives the message M_2 . If it is, the remote server accepts the login request; otherwise it refuses the login. The figure of our improvement on Yang and Wang's nonce-based password authentication scheme is shown in Figure 11.

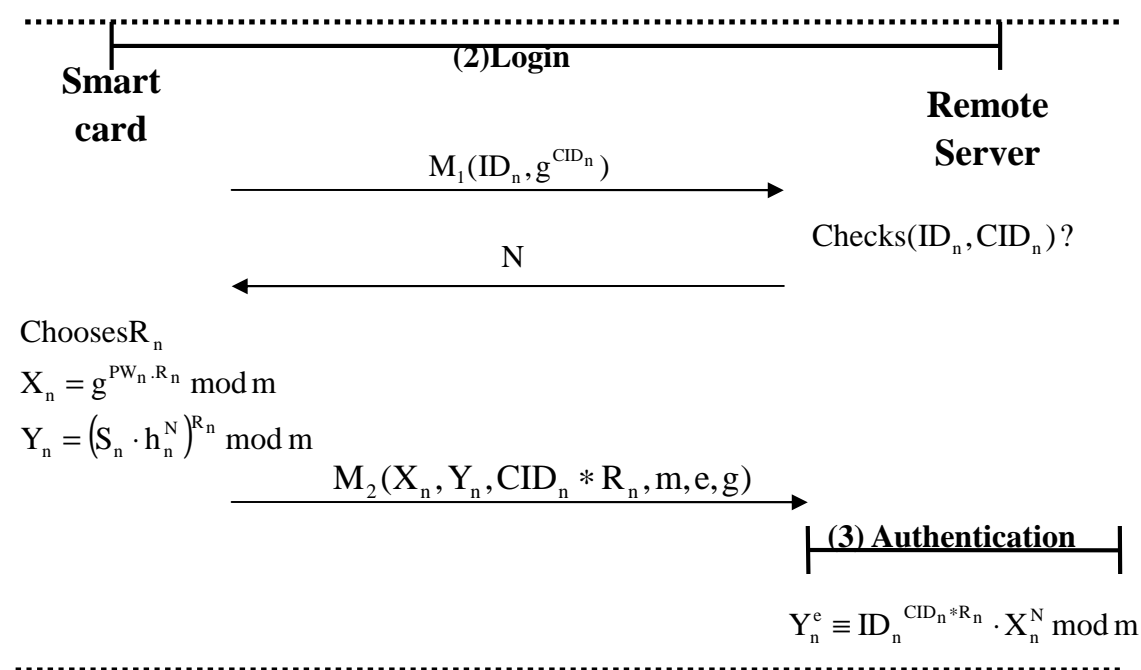


Figure 11: Our improvement on Yang and Wang's nonce-based password authentication scheme

4.2 Security analysis

In this section, we will analyze some familiar attacks on our improvements. They are forgery attacks, password-guessing attack, replay attack and smart card loss. All of them are described as follows.

4.2.1 Forgery attack

In the timestamp-based password authentication scheme, an attacker can get the $(ID_n, g^{CID_n}, CID_n * R_n, X_n, Y_n, m, e, g, T)$ by eavesdropping on the communication line. But it is impossible for him to know the CID_n from the g^{CID_n} or $CID_n * R_n$. Since we know computing the CID_n from the value of g^{CID_n} is a discrete logarithm problem. Besides, because of the attacker does not know the value R_n , he cannot compute the value CID_n from $CID_n * R_n$ as well. Since the attacker can't know both the CID_n and R_n , he can't know the value of $ID_n^{CID_n * R_n}$ in the verifying equation $Y_n^e \equiv ID_n^{CID_n * R_n} \cdot X_n^T \pmod{m}$ and thus he can not forge the value of X_n, Y_n to satisfy the equation $\frac{Y_n^e}{X_n^T} \equiv ID_n^{CID_n * R_n} \pmod{m}$. Therefore the forgery attack cannot succeed in the timestamp-based password authentication scheme.

In the nonce-based password authentication scheme, an attacker also can get $(ID_n, g^{CID_n}, CID_n * R_n, X_n, Y_n, m, e, g)$ by intercepting the communication message and N for the corresponding (ID_n, CID_n) from step2 in the login phase. But the forgery attack can't work successfully in this nonce-based password authentication scheme on the basis of above explained reason.

4.2.2 Password-guessing attack

An attacker has two ways to obtain the password PW_n , one is from

$h_n = g^{\text{PW}_n \cdot d} \bmod m$ in the smart card, and the other is from $X_n = g^{\text{PW}_n \cdot R_n} \bmod m$ in the login phase of both the timestamp-based and nonce-based password authentication schemes. However, because the attacker cannot know the values of d and R_n due to the intractability of the discrete log problem in Z_n , he can't get the password PW_n from the two equations. Therefore, our improvements can withstand the password-guessing attack.

4.2.3 Replay attack

If an attacker wants to replay the message M that consists of $(\text{ID}_n, g^{\text{CID}_n}, \text{CID}_n * R_n, X_n, Y_n, m, e, g, T)$ to the remote server in both the timestamp-based and nonce-based password authentication schemes. The remote server will refuse it after receiving the replay message. Since, first, the replay message might not be able to pass the verification $T' - T \leq \Delta T$ in the step2 of the authentication phase. Second, when the server sees the same value $\text{CID}_n * R_n$ as used before, it knows this is a replay attack. Because of the value $\text{CID}_n * R_n$ should be different from the ever used ones due to the random value R_n each time.

4.2.4 Smart card loss

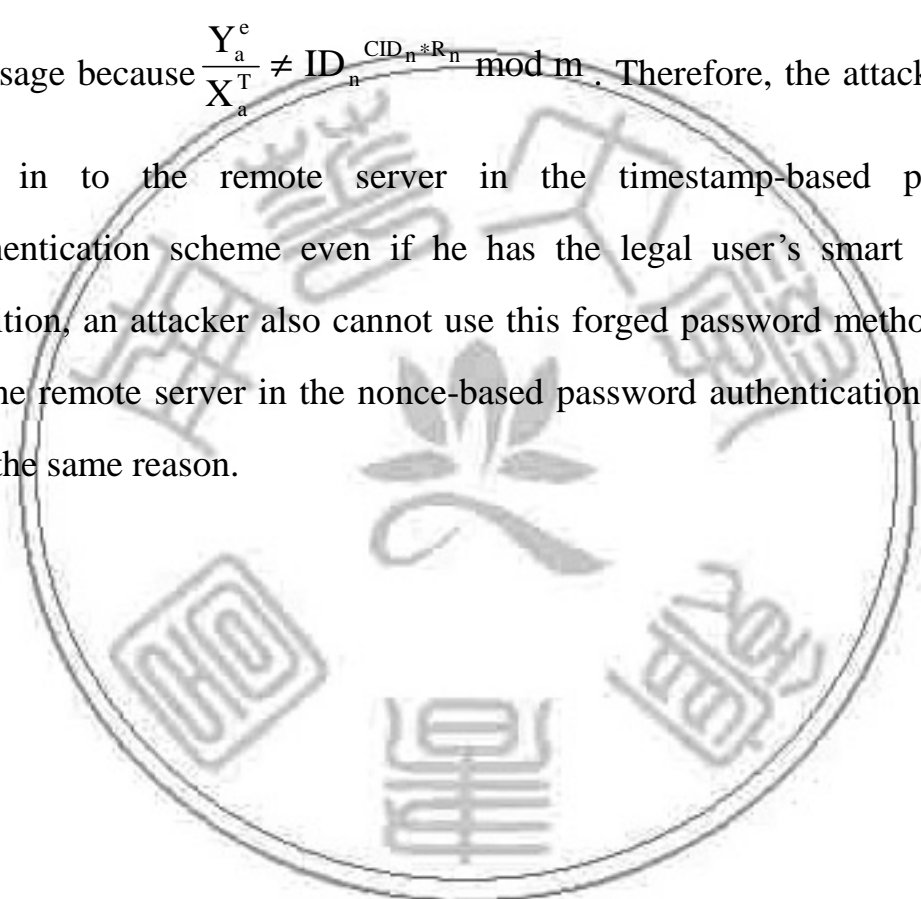
If a legal user loses his smart card and it might happen to be discovered by an attacker. Then, due to the difficulties of password-guessing attack explained above, the attacker can't guess the legal user's password successfully. Even if the attacker could insert the

smart card into the input device and key in his forged password PW_a and to impersonate U_n . Then the smart card computes $X_a = g^{PW_a \cdot R_n} \bmod m$, $Y_a = (S_n \cdot h_n^T)^{R_n} \bmod m$ and sends the login message to the remote server.

But this login message will not be able to pass the verifying equation $Y_n^e \equiv ID_n^{CID_n \cdot R_n} \cdot X_n^T \bmod m$ after the remote server receives the

message because $\frac{Y_a^e}{X_a^T} \neq ID_n^{CID_n \cdot R_n} \bmod m$. Therefore, the attacker can't

log in to the remote server in the timestamp-based password authentication scheme even if he has the legal user's smart card. In addition, an attacker also cannot use this forged password method to log to the remote server in the nonce-based password authentication scheme for the same reason.



Chapter 5 Conclusion

In this thesis we have pointed out the weaknesses existed in both of Yang and Wang's password authentication schemes and Boyd and Mao's Deniable Authenticated key Establishment Protocols. The weaknesses of Yang and Wang's password authentication schemes are forgery attack that an attacker can masquerade as a legal user and cheat the remote server successfully. Similarly the vulnerability of Boyd and Mao's Deniable Authenticated key Establishment Protocols are KCI attack that if A's long-term secret key is compromised and known by an adversary, the adversary can pretend others to communicate with A. Therefore, they are not secure enough on the execution of their protocol. Then we also have improved the weaknesses of Yang and Wang's password authentication schemes by our proposals. After a deeper security analysis as shown in Section 4.2, we have showed that our improvement schemes are secure. Therefore, after our enhancement, the strength of Yang and Wang's password authentication schemes is equivalent to the intractability of the discrete log problem

References

- [1] A.J. Menezes and T. Okamoto, “Reducing elliptic curve logarithms to a Finite Field,” *IEEE Transaction on Information Theory*, Vol.39, No.5, pp. 1639-1646, 1993.
- [2] C. C. Lee and M.S. Hwang, W.P. Yang, “A flexible remote user authentication scheme using smart cards,” *ACM Operating Systems Review* Vol.36, No.3, pp. 46-52, 2002.
- [3] C. C. Yang and R. C. Wang, “An improvement of the Yang-Shieh password authentication schemes,” *Applied Mathematics and Computation*, pp. 1391-1396, 2005.
- [4] C. K. Chan and L. M. Cheng, “Cryptanalysis of a timestamp based password authentication scheme,” *Computers & Security*, Vol.21, No.1, pp. 74-76, 2002.
- [5] D. Boneh and M. Franklin, “Identity-based encryption from the Weil pairing,” In *Cryptology-CRYPTO 2001*, 21st Annual International Cryptology Conference, volume 2139 of LNCS, pp. 213-229, 2001.
- [6] D. Harkins and D. Carrel, “The Internet Key Exchange (IKE),” *Internet RFC 2409*, 1998.
- [7] F. Zhang and S.N. Reihaneh, “ID-Based Chameleon Hashes from Bilinear Pairings,” *Cryptology ePrint Archive: Report*, No.128, 2003.
- [8] G. Boyd and W. Mao, “Deniable Authenticated Key Establishment for Internet Protocols,” *Security Protocols workshop of Cambridge, UK*, pp. 255-271, (2003).
- [9] H.M. Sun and B.T. Hsieh, “Security Analysis of Shim’s Authenticated Key Agreement Protocols from Pairings,” *Cryptology ePrint Archive: Report*, No.113, 2003.
- [10] H. M. Sun and H.T. Yeh, “Further cryptanalysis of a password authentication scheme with smart cards,” *IEICE Transactions and Communications* Vol.E86-B, No.4, pp. 1412-1415, 2003.
- [11] H.S. Lee, “A self-pairing map and its applications to cryptography,” *Applied Mathematics and Computation*, Vol.151, pp. 671-678, 2004.

- [12]I. Duursma and H.S. Lee, “A group key agreement protocol from pairings,” *Applied Mathematics and Computation*, Vol.167, pp. 1451-1456, 2005.
- [13]J. J. Shen and C. W. Lin, “Security enhancement for the timestamp based password authentication scheme using smart cards,” *Computers & Security*, Vol.22, No.7, pp. 591-595, 2003.
- [14]J. Zhou, “ Further analysis of the Internet key exchange protocol, ” *Computer Communications*, Vol.23, pp. 1606-1612, 2000.
- [15]K. F. Chen and S. Zhong, “Attacks on the Yang-Shieh authentication,” *Computers & Security*, Vol.22, No.8, pp. 725-727, 2003.
- [16]M. Bellare and P. Rogaway, “ Provably secure session key distribution -the three party case, ” In *Proceedings of the 27th ACM Symposium on the Theory of Computing*, 1995.
- [17]M.S. Borella, “ Methods and protocols for secure key negotiation using IKE, ” *IEEE Network*, pp. 18-29, 2000.
- [18]R. Canetti and H. Krawczyk, “ Security analysis of IKE's signature-based key exchange protocol, ” *Proceedings of the 22nd Annual International Cryptology Conference on Advances in Cryptology*, 2002.
- [19]R. Jiang and Li. Pan, “Further analysis of password authentication schemes -based on authentication tests, ” *Computers & Security*, pp. 469-477, 2004.
- [20]R. Lu and Z. Cao, “ A new deniable authentication protocol from bilinear pairings, ” *Applied Mathematics and Computation*, Vol.168, pp. 954-961, 2005.
- [21]R. Perlman and C. Kaufman, “ Key exchange in IPSec: Analysis of IKE, ” *IEEE Internet Computing*, pp. 50-56, 2000.
- [22]R. Sakai and K. Ohgishian, “ Cryptosystems based on pairing, ” In *The 2000 Symposium on Cryptography and Information Security*, Okinawa, Japan, 2000.
- [23]S. B. Wilson, and A. Menezes, “Authenticated Diffie-Hellman key agreement protocols, ” *Proceedings of the 5th Annual Workshop on*

Selected Areas in Cryptography (SAC '98), Lecture Notes in Computer Science, pp. 339-361, 1999.

- [24]S. T. Wu and B.C. Chieu, "A user friendly remote authentication scheme with smart cards, " Computers & Security, Vol.22, No.6, pp. 547-550, 2003.
- [25]W. H. Yang and S. P. Shieh, "Password authentication schemes with smart cards, " Computers & Security, Vol.18, No.8, pp. 727-733, 1999.
- [26]W. J. Tsaur and C.C. Wu, "A smart card-based remote scheme for password authentication in multi-server Internet services, " Computer Standards & Interfaces, Vol.27, pp. 39-51, 2004.
- [27]W. S. Juang, "Efficient password authenticated key agreement using smart card, "Computers & Security, Vol.23, pp. 167-173, 2004.
- [28]Y.J. Choie and E.J. Jeong, "Efficient identity-based authenticated key agreement protocol from pairings, " Applied Mathematics and Computation, Vol.162, pp. 179-188, 2005.
- [29]Z. Chen, "Security analysis on Nalla-Reddy's ID-based tripartite authenticate key agreement protocols, " Cryptology ePrint Archive: Report, No.103, 2003.
- [30]Z.F. Zhang and J. XU, "Attack on an Identification Scheme Based on Gap Diffie-Hellman Problem, " Cryptology ePrint Archive: Report, No.153, 2003.