Cryptanalysis of Timestamp-Based Password Authentication Schemes Using Smart Cards

Guilin Wang and Feng Bao

Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
{glwang, baofeng}@i2r.a-star.edu.sg

Abstract. Password authentication is an important mechanism for remote login systems, where only authorized users can be authenticated via using their passwords and/or some similar secrets. In 1999, Yang and Shieh [14] proposed two password authentication schemes using smart cards. Their schemes are not only very efficient, but also allow users to change their passwords freely and the server has no need to maintain a verification table for authenticating users. However, their schemes are later identified to be flawed. To overcome those security flaws, Shen et al. [9] and Yoon et al. [17] proposed further improvements and claimed their new schemes are secure. In this paper, we first point out that Yang et al.'s attack [15] against Shen et al.'s scheme is actually *invalid*, since we can show that in a real implementation it is extremely difficult to find two hash values such that one is divisible by the other. After that, we show that both of Shen et al.' scheme and Yoon et al.'s scheme are *insecure* by identifying several effective impersonation attacks. Those attacks enable an outsider to be successfully authenticated and then enjoy the resources and/or services provided by the server.

Keywords: password authentication, smart card, attack, hash function.

1 Introduction

Password authentication is an important mechanism for remote login systems to implement remote authentication through a public and insecure network, such as Internet. In such a system, it is required that only authorized users can be authenticated by the server, and then are granted to access the resources and/or services provided by the server. Since in this environment users usually hold portable but capability-limited devices such as smart cards with passwords, it is highly desirable that only simple and efficient operations, rather than complicated cryptographic techniques, are exploited to implement the authentication procedure.

The first remote authentication scheme is proposed by Lamport in 1981 [6]. After that, a number of password authentication schemes [14, 5, 3, 4, 9] have been proposed and analyzed due to the facts that those schemes are both potentially important in practical applications and amazingly attractive in their simple structures. In 1999, Yang and Shieh [14] proposed two password authentication schemes using smart cards, one is timestamp-based and the other is

P. Ning, S. Qing, and N. Li (Eds.): ICICS 2006, LNCS 4307, pp. 399–409, 2006.

nonce-based. Compared with previous schemes, their schemes are very interesting since the following two features are achieved: (a) All legal users are allowed to set and change their passwords freely; and (b) The server has no need to maintain a verification table for authenticating users. Later, Chan and Cheng [3], and Fan et al. [4] pointed out that the Yang-Shieh scheme is vulnerable to impersonation attacks if the users' identities IDs are not carefully formatted or encoded. To thwart those attacks, Shen, Lin, and Hwang [9] improved the Yang-Shieh scheme. However, Yang, Yang and Wang [15] recently presented a simple attack against Shen at al.'s scheme by finding two hash values so that one is a multiple of the other. In another direction, Yang, Wang, and Chang [16] also enhanced the Yang-Shieh scheme. However Yoon et al. [17] showed that Yang et al.'s schemes in [16] are also insecure and further proposed improvements.

In this paper, we present a cryptanalysis of two above mentioned timestamp-based password authentication schemes, i.e., the SLH scheme [9] and the YKY scheme [17]. We show that both of those two authentication schemes are vulnerable to impersonation attacks, and that the YYW attack [15] against the SLH scheme is invalid. In more detail, this paper has the following contributions. We first point out that the YYW attack against the SLH scheme is actually *invalid*, since we can show that in a real implementation it is *extremely difficult* to find two hash values such that one is divisible by the other. Precisely speaking, we prove that if a hash function is modelled as a random function [1], then the probability that one hash value is divisible by another is less than $(1+k)/2^k$, where k is the fixed output length of the hash function. In real applications, however, k should be set as 128 at least. Then, on the other hand, we show that Shen et al.'s password authentication scheme is indeed *insecure* by successfully identifying two effective impersonation attacks. By exploiting our attacks, the attacker as an outsider can impersonate a legitimate user to access the resources and/or services provided by the server. Finally, we demonstrate that the YKY scheme is also insecure, contrary to the authors' claim in [17], since it suffers a similar impersonation attack.

The rest of the paper is organized as follows. We first review the SLH scheme in Section 2, discuss the invalidity of the YYW attack in Section 3, and present our impersonation attacks against the SLH scheme in Section 4. Then, we turn to review and analyze the YKY scheme in Section 5. Finally, the conclusion is given in Section 6.

2 Review of the SLH Scheme

The SLH timestamp-based password authentication scheme [9] consists of three phases: registration, login, and authentication. We now review each phase as follows.

2.1 Registration Phase

It is assumed that the server has an RSA cryptosystem [8] with key material (n, e, d), where $n = pq$ is the product of two large primes p and q, e is a prime number, and $d = e^{-1} \bmod (p-1)(q-1)$. Furthermore, g is a primitive element in both $GF(p)$ and $GF(q)$, and $f(\cdot)$ is a secure hash function. Except (d, p, q) are kept as secrets, $(n, e, g, f(\cdot))$ are publicly published.

When a user U_i wants to register with the server, he/she first submits his/her identity ID_i and a chosen password PW_i to the server through a secure channel. Then, the server computes three values of (S_i, h_i, CID_i) as follows:

$$S_i = ID_i^d \bmod n, \quad h_i = g^{PW_i \cdot d} \bmod n, \text{ and } CID_i = f(ID_i \oplus d), \tag{1}$$

where $\oplus$ denotes the exclusive operation, and CID_i is treated as the card identity.

Finally, the server writes $(n, e, g, f, ID_i, CID_i, S_i, h_i)$ into a smart card, and then delivers this smart card to the user U_i.

2.2 Login Phase

When the user U_i wants to login into the server, he/she inserts his/her smart card into a card reader and enters his/her identity ID_i and password PW_i. If both ID_i and PW_i are valid, the smart card selects a random number r_i, and then computes values X_i and Y_i by[1]

$$X_i = g^{r_i \cdot PW_i} \bmod n \text{ and } Y_i = S_i \cdot h_i^{r_i \cdot f(CID_i, T_1)} \bmod n, \tag{2}$$

where the timestamp T_1 denotes the current date and time when this login occurs. Finally, the login request message $M = \{ID_i, CID_i, X_i, Y_i, n, e, g, T_1\}$ is sent to the server.

2.3 Authentication Phase

Upon receiving $M = \{ID_i, CID_i, X_i, Y_i, n, e, g, T_1\}$, the server checks the validity of this login request message according to the following procedures:

- $CID_i \equiv f(ID_i \oplus d)$.
- $Y_i^e \equiv ID_i \cdot X_i^{f(CID_i, T_1)} \bmod n$.
- $T_2 - T_1 \leq \Delta T$, where T_2 denotes the date and time when the server received the request M, and ΔT is a predefined time interval to balance the reasonable transmission delay and potential replay attack.

If any of the above verifications fails, the login request is rejected. Otherwise, the server first calculates $R = f(CID_i, T_2)^d \bmod n$, and then sends back message $N = \{R, T_2\}$ to the user U_i. After receiving message N, the user U_i accepts the server's service if and only if both of the following checks hold:

- $R^e \equiv f(CID_i, T_2) \bmod n$.
- $T_3 - T_2 \leq \Delta T$, where T_3 denotes the date and time when U_i received message N.

[1] Note that in [15], the value X_i is calculated by $X_i = g^{r_i \cdot f(CID_i, T_1)} \bmod n$. However, this formula is incorrect since it is inconsistent with the original specification of the SLH scheme [9]. Therefore, this typo is corrected here.

3 The YYW Attack and Its Invalidity

Under the assumption that an attacker can find two hash values such that one is a multiple of the other, Yang et al. [15] identified the following attack on the SLH scheme.

1. The attacker first intercepts a login request message $M = \{ID_i, CID_i, X_i, Y_i, n, e, g, T_1\}$ over the communication channel.
2. Then, the attacker finds a value a such that $a \cdot f(CID_i, T_1') = f(CID_i, T_1)$, where T_1' is the attacker's login time.
3. Finally, the attacker sends a forged login request message $M' = \{ID_i, CID_i, X_i', Y_i, n, e, g, T_1'\}$ to the server, where X_i' is computed by

$$X_i' = X_i^a \bmod n \; (= g^{r_i \cdot PW_i \cdot a} \bmod n).$$

It is easy to know that this attack is correct, since both (ID_i, CID_i) and (X_i', Y_i) are valid pairs, i.e.,

$$CID_i = f(CD_i \oplus d) \text{ and } Y_i^e = ID_i \cdot X_i'^{f(CID_i, T_1')} \bmod n.$$

However, we notice that the YYW attack is actually *invalid* in practice, because it is *extremely difficult* to find a login time T_1' such that $f(CID_i, T_1')$ is a factor of $f(CID_i, T_1)$. Formally, we have the following theorem.

Theorem 1. *Let X and Y be two random inputs of a hash function $f(\cdot)$ with k-bit output. Then, under the assumption that the outputs of hash function $f(\cdot)$ can be considered as random numbers, the probability that $f(Y)$ is divisible by $f(X)$ is at most $(k+1)/2^k$. That is, we have*

$$P \triangleq Pr[f(X)|f(Y)] \leq (1+k)/2^k. \tag{3}$$

Proof: Let $x = f(X)$ and $y = f(Y)$. Since the outputs of hash function $f(\cdot)$ are assumed to be random numbers with k bits, x and y can be treated as two random integers independently chosen from interval $[0, 2^k - 1]$. More specifically, pair (x, y) could be any element of set $S = \{(a, b)|\forall a, b \in [0, 2^k - 1]\}$ with equal probability 2^{-2k}.

To compute probability P, we need to count how many pairs (a, b) in set S satisfying $a|b$, i.e., there exists an integer t such that $b = a \cdot t$. That is, we have to compute or estimate the cardinality of subset $T = \{(a, b)|(a, b) \in S \wedge a|b\}$. Actually, we can estimate $|T|$, i.e., the numbers of pairs in subset T, according to the value of a as follows:

- $a = 0$: there is only one pair in T, i.e., $(0, 0)$;
- $a = 1$: there are 2^k pairs in T, i.e., all $(1, b)$ for any $b \in [0, 2^k - 1]$;
- $a = 2$: there are at most $(1 + \frac{2^k}{2})$ pairs in T;
- $a = 3$: there are at most $(1 + \frac{2^k}{3})$ pairs in T;
- ......;

- $a = i$: there are at most $(1 + \frac{2^k}{i})$ pairs in T;
- ......;
- $a = 2^k - 1$: there are at most $(1 + \frac{2^k}{2^k-1})$ pairs in T.

Therefore, we have the following estimate for the upper bound of $|T|$:

$$\begin{aligned}|T| &\leq 1 + 2^k + (1 + \tfrac{2^k}{2}) + (1 + \tfrac{2^k}{3}) + \cdots + (1 + \tfrac{2^k}{i}) \\ &\quad + \cdots + (1 + \tfrac{2^k}{2^k-1}) \\ &\leq 2^k + 2^k + \tfrac{2^k}{2} + \tfrac{2^k}{3} + \cdots + \tfrac{2^k}{i} + \cdots + \tfrac{2^k}{2^k-1} \\ &\leq 2^k \cdot [2 + (\tfrac{1}{2} + \tfrac{1}{3}) + (\tfrac{1}{4} + \cdots + \tfrac{1}{7}) + \cdots + (\tfrac{1}{2^j} + \\ &\quad \cdots + \tfrac{1}{2^{j+1}-1}) + \cdots + (\tfrac{1}{2^{k-1}} + \cdots + \tfrac{1}{2^k-1})] \\ &\leq 2^k[2 + \textstyle\sum_{j=1}^{k-1} 2^j \cdot \tfrac{1}{2^j}] \\ &\leq 2^k[1+k].\end{aligned}$$

As $|S| = 2^{2k}$, we consequently get the following upper bound for probability P:

$$P = Pr[f(X)|f(Y)] = |T|/|S| \leq (1+k)/2^k. \tag{4}$$

This is what we want to prove. □

In a real system, the output length of hash function $f(\cdot)$ is at least 128-bit. In this case ($|f(\cdot)| = 128$), according to Proposition 1 we know that that for any randomly chosen T_1', the probability that $f(CID_i, T_1)$ is a multiple of $f(CID_i, T_1')$ is at most $(1+128)/2^{128} < 2^{-120}$, a negligible quantity. As specified in Proposition 1, this statement holds under the assumption that the outputs of hash function $f(\cdot)$ can be treated as random integers with fixed length [1]. Naturally, a real-world hash function cannot be *completely* treated as a random function. However, as one of cryptographic requirements on hash functions they should be *very* close to a random function. In fact, this treatment is a popular method exploited in modern cryptography research, called random oracle model, first introduced by Bellare and Rogaway in [1].

Note that just due to the fact that probability $P = (1+k)/2^k$ is negligible in security parameter k, i.e., the fixed output length of hash function $f(\cdot)$, an attacker can neither run the YYW attack by polynomial times (in k) to get an successful login with a non-negligible probability. For example, in the case $k = 128$, to get a successful login by using the YYW attack the attacker have to try about 2^{120} times. If one try needs one second to finish, this means the attacker have to cost over than 2^{95} years to succeed one impersonating login. This is the exact reason why we say the YYW attack is invalid or infeasible in practice. However, as shown in next section, the SLH authentication scheme is truly weak and can be attacked by an outsider without much cost.

4 New Attacks Against the SLH Scheme

In this section, we show that the SLH authentication scheme [9] is indeed insecure by presenting two effective attacks, though the YYW attack is invalid as we just discussed. The first attack can be mounted if the RSA public exponent e is a small prime number, while the second attack works without any assumption on the size of public RSA exponent e.

4.1 Impersonation Attack A

The SLH authentication scheme [9] just requires the RSA public exponent e should be a prime number but did not specify the size of e. In other words, one may implement the SLH authentication scheme by selecting a small prime number as the value of e, for example, 3, 7, 13, 17 etc. Actually, this is likely to happen due to two reasons: (1) Some standards, e. g. PKCS #1 [7], recommend to use small exponent e such as 3 to speed up the RSA signature verification; and (2) Small exponent e can reduce the computational cost of smart cards, which are employed in the SLH scheme as the authentication devices for users.

However, if the exponent e is truly set as a small prime number, the SLH authentication scheme is vulnerable to the following impersonation attack A.

1. The attacker first intercepts a login request message $M = \{ID_i, CID_i, X_i, Y_i, n, e, g, T_1\}$ over the communication channel.
2. Then, the attacker checks whether $f(CID_i, T_1)$ is divisible by e or not, i.e., $e|f(CID_i, T_1)$. If not, intercept more login request messages. Otherwise, continue.
3. Let $f(CID_i, T_1) = eb$ for some integer $b \in Z$. Then, compute S_i by
$$S_i = Y_i \cdot X_i^{-b} \bmod n. \tag{5}$$
4. For any timestamp T_1', the attacker selects a random number $r \in Z_n$, and then compute X_i' and Y_i' as follows:
$$X_i' = r^e \bmod n \text{ and } Y_i' = S_i \cdot r^{f(CID_i, T_1')} \bmod n. \tag{6}$$
5. Finally, the attacker can impersonate user U_i to access the server by sending out a forged login request message $M' = \{ID_i, CID_i, X_i', Y_i', n, e, g, T_1'\}$.

Note that in the above attack, we have $CID_i \equiv f(ID_i \oplus d)$ and $Y_i'^e \equiv ID_i \cdot X_i'^{f(CID_i, T_1')} \bmod n$. The latter formula is justified by the following equalities:

$$\begin{aligned} Y_i'^e &= [S_i \cdot r^{f(CID_i, T_1')}]^e \bmod n \\ &= S_i^e \cdot (r^e)^{f(CID_i, T_1')} \bmod n \\ &= (Y_i \cdot X_i^{-b})^e \cdot X_i'^{f(CID_i, T_1')} \bmod n \\ &= (Y_i^e \cdot X_i^{-be}) \cdot X_i'^{f(CID_i, T_1')} \bmod n \\ &= (Y_i^e \cdot X_i^{-f(CID_i, T_1)}) \cdot X_i'^{f(CID_i, T_1')} \bmod n \\ &= ID_i \cdot X_i'^{f(CID_i, T_1')} \bmod n. \end{aligned} \tag{7}$$

Therefore, our attack A is successful if the forged login request message $M' = \{ID_i, CID_i, X_i', Y_i', n, e, g, T_1'\}$ can be delivered to the server before $T_1' + \Delta T$. This is not a problem for the attacker, since this condition applies to all legal users too. The only concern is the probability of $e|f(CID_i, T_1)$. Under the assumption that the outputs of hash function $f(\cdot)$ are random numbers, it is easy to know $e|f(CID_i, T_1)$ for a random timestamp T_1 with probability of $1/e$. This implies that to successfully amount attack A, the attacker only needs to intercept a dozen of valid login messages on average if e is an odd prime less than 20. Actually, even if $e = 65537 = 2^{16} + 1$ attack A remains feasible in practice if an attacker eavesdrops thousands of valid login messages (not limited to one single legitimate user).

However, note that if $|e| \geq 80$ it seems infeasible to amount attack A since each single run of the attacking algorithm with success probability only about 2^{-80}, a negligible quantity. This is the reason why we have to assume that e should be a small number in attack A. However, attack B described in the next section does not rely on this assumption any more.

4.2 Impersonation Attack B

In this attack, to access the server by impersonating the user U_i an attacker as outsider just needs to know user U_i's identity ID_i and smart card identity CID_i. That is, to mount our attack it is sufficient to intercept one valid login request message $M = \{ID_i, CID_i, X_i, Y_i, n, e, g, T_1\}$, made by user U_i. After that, to login the server at timestamp T_1' the attacker checks whether $\gcd(e, f(CID_i, T_1')) = 1$, i.e., whether the integers e and $f(CID_i, T_1)$ are relatively prime to each other. Note that for two randomly selected integers u and v, $\gcd(u, v) = 1$ happens with probability $6/\pi^2 \approx 0.6$ [10, 11]. Since the hash function $f(\cdot)$ is usually considered as a random function with k-bit outputs, $\gcd(e, f(CID_i, T_1')) = 1$ should occur with a similar probability. However, due to the fact that the RSA public exponent e is required to be a prime number in the SLH scheme, it is not difficult to see that for arbitrary timestamp T_1' $\gcd(e, f(CID_i, T_1')) = 1$ holds with probability $1 - 1/e$, if the outputs of hash function $f(\cdot)$ are assumed to be random numbers with k bits. Therefore, to get a value $f(CID_i, T_1')$ such that $\gcd(e, f(CID_i, T_1')) = 1$, the attacker *only* needs to try one or two timestamps. Once such a timestamp T_1' is obtained, the attacker can complete the following impersonation attack B:

1. Since $\gcd(e, f(CID_i, T_1')) = 1$, the attacker can use the Extended Euclidean algorithm to compute two integers a and b such that

$$a \cdot e + b \cdot f(CID_i, T_1') = 1 \quad (\text{in } \mathbb{Z}). \tag{8}$$

2. Then, the attacker computes X_i' and Y_i' by

$$X_i' = (ID_i)^{-b} \bmod n, \quad Y_i' = (ID_i)^a \bmod n. \tag{9}$$

3. Finally, the attacker sends the forged login request message $M' = \{ID_i, CID_i, X_i', Y_i', n, e, g, T_1'\}$ to the server.

Again, the above attack is successful since we have $CID_i \equiv f(ID_i \oplus d)$ and $Y_i'^e \equiv ID_i \cdot X_i'^{f(CID_i,T_1')} \bmod n$. The latter expression is justified as follows:

$$\begin{aligned} Y_i'^e &= [(ID_i)^a]^e \bmod n \\ &= ID_i^{ae} \bmod n \\ &= ID^{1-b\cdot f(CID_i,T_1')} \bmod n \\ &= ID_i \cdot (ID_i{}^{-b})^{f(CID_i,T_1')} \bmod n \\ &= ID_i \cdot X_i'^{f(CID_i,T_1')} \bmod n. \end{aligned} \tag{10}$$

Remark 1: Note that neither of the attacks from Chan and Cheng [3] and Fan et al. [4] can apply to the SLH scheme, since in the SLH scheme user's identity ID_i is validated by checking $CID_i \equiv f(ID_i \oplus d)$. Therefore, without the secret d anybody cannot forge a valid smart card identity CID_i for an identity ID_i. In addition, it is also infeasible to derive the secret d from $CID_i = f(ID_i \oplus d)$ via off-line attacks, since d should be a large number [2]. In the case of $|n| = 1024$, this means we are supposed to select d such that $|d| \geq 300$.

5 The YKY Scheme and Its Security

Since the YKY authentication scheme [17] is also an enhancement of the Yang-Shieh scheme [14], it has a similar structure as the SLH scheme [9]. In this section, we briefly overview the YKY scheme and analyze its security.

5.1 Review of the YKY Scheme

The three phases of the YKY scheme are recalled as follows.

1. Registration Phase: As in the SLH scheme, the server sets an RSA cryptosystem with key material (n, e, d), where $n = pq$ and $ed = 1 \bmod (p-1)(q-1))$, and makes $(n, e, g, f(\cdot))$ public, while keeping d secret. To be registered, a user U_i securely delivers his/her identity ID_i and a chosen password PW_i to the server. After that, the server issues user U_i a smart card which contains information $(n, e, g, f(\cdot), ID_i, CID_i, S_i^*, h_i)$, where

$$S_i^* = ID_i^{CID_i \cdot d} \bmod n, \quad h_i = g^{PW_i \cdot d} \bmod n, \text{ and } CID_i = f(ID_i \oplus d). \tag{11}$$

2. Login Phase: To access the server, user U_i inserts his/her smart card into a card reader and types the password PW_i. If the password PW_i is correct, the smart card sends a login request message $M = \{ID_i, CID_i^*, X_i, Y_i^*, n, e, g, T_1\}$ to the server by computing

$$CID_i^* = CID_i^e \bmod n, \ X_i = g^{r_i \cdot PW_i} \bmod n, \text{ and } Y_i^* = S_i^* \cdot h_i^{r_i \cdot T_1} \bmod n. \tag{12}$$

Here, r_i is a randomly chosen number and T_1 is the current date and time.

3. Authentication Phase: Once M is received, the server accepts user U_i's login request if and only if all of the following verifications hold:

- Check the validity of ID_i.
- Check $T_2 - T_1 \leq \Delta T$, where T_2 denotes the date and time when the server received M, and ΔT is a appropriately predefined time interval.
- Compute $CID_i = (CID_i^*)^d \bmod n$, and check that $CID_i \equiv f(ID_i \oplus d)$.
- Check $(Y_i^*)^e \equiv ID_i^{CID_i} \cdot X_i^{T_1} \bmod n$.

In contrast, the values of (S_i^*, CID_i^*, Y_i^*) in the YKY scheme are used to replace (S_i, CID_i, Y_i) in the SLH scheme. Especially, the smart card identifier CID_i contained in M is transferred as a ciphertext CID_i^* rather than plaintext. This reason is that by using a valid smart card identifier CID_i, Yoon et al. [17] launched an impersonation attack against the YWC scheme [16]. So, in their improvemed YKY scheme CID_i is not transferred in plaintext anymore. In addition, note that the YKY scheme only enables the server to authenticate a user, while the SLH scheme provides both directions of authentication service.

5.2 Security of the YKY Scheme

In [17], Yoon et al. claimed that the YKY scheme can resist impersonation attack, password guessing attack, smart card loss attack, and replay attack. They argued that their scheme is immune to impersonation attack, since an attacker without the server's secret d cannot derive CID_i from its RSA ciphertext CID_i^*. Without the card identifier CID_i, however, the attacker cannot forge a pair (X_i', Y_i') such that $(Y_i')^e \equiv ID_i^{CID_i} \cdot X_i'^{T_1} \bmod n$.

We notice that to amount a personation attack in the YKY scheme, an attacker *does not* need to get the value of CID_i at all. The attacking strategy is analogous to Attack B against the SLH scheme. To this end, an attacker first intercepts a valid login request message $M = \{ID_i, CID_i^*, X_i, Y_i^*, n, e, g, T_1\}$, which is sent to the server by some legitimate user U_i. Due to the validity of M, we have $(Y_i^*)^e \equiv ID_i^{CID_i} \cdot X_i^{T_1} \bmod n$. Therefore, the attacker gets the following value A by computing

$$A = (Y_i^*)^e \cdot X_i^{-T_1} \bmod n \ (= ID_i^{CID_i} \bmod n). \tag{13}$$

After that, the attacker computes a timestamp T_1' such that $\gcd(e, T_1') = 1$ (This even happens with probability about $1 - 1/e$, as e is a prime). So, the attacker can use the Extended Euclidean algorithm to compute two integers a and b such that

$$a \cdot e + b \cdot T_1' = 1 \quad (\text{in } \mathbb{Z}). \tag{14}$$

Finally, the attacker sends the forged login request message $M' = \{ID_i, CID_i^*, X_i', Y_i', n, e, g, T_1'\}$ to the server, where

$$X_i' = A^{-b} \bmod n \ \text{ and } \ Y_i' = A^a \bmod n. \tag{15}$$

It is easy to know that the above attack is successful, since we have $CID_i \equiv f(ID_i \oplus d) \equiv (CID_i^*)^e \bmod n$ and $Y_i'^e \equiv ID_i^{CID_i} \cdot X_i'^{T_1'} \bmod n$. The latter expression is justified by

$$Y_i'^e = A^{ae} \bmod n = A^{1-b \cdot T_1'} \bmod n = A \cdot (A^{-b})^{T_1'} \bmod n = ID_i^{CID_i} \cdot X_i'^{T_1'} \bmod n.$$

Remark 2: Interestingly, we note that similar impersonation attack cannot apply to Yoon et al.'s nonce-based password authentication scheme (See Section 4.2 of [17]). In this scheme, to access the server a user U_i needs to send a request message $M = (X_i, Y_i, n, e, g)$ such that $Y_i^e \equiv ID_i^{CID_i} \cdot X_i^N \bmod n$, where $N = f(CID_i, r_j)$ and r_j is a random number selected by the server. Since both values of CID_i and N are unavailable to an attacker, it seems really hard to amount an impersonation attack.

6 Conclusion

Password authentication is an important mechanism for remote login systems that enables the server to authenticate its users. In this paper, we first pointed out that Yang et al.'s attack [15] against Shen at al.'s timestamp-based password authentication scheme [9] is actually invalid, since we showed that in a real implementation it is extremely difficult to find two hash values such that one is divisible by the other. Then, we showed that Shen et al.'s authentication scheme is really *insecure* by demonstrating two effective impersonation attacks. Finally, we illustrated that Yoon et al.'s timestamp-based authentication scheme [17] is also suffers to a similar personation attack. In our security analysis, we employed the following two facts on hash functions: (1) If the outputs of a hash function can be modelled as random numbers with fixed length k, the probability that one hash value is a multiple of another is less than $(1+k)/2^k$, a negligible quantity in k; and (2) The probability that one hash value is relatively prime with another hash value (or a fixed integer), however, is certainly high, about 0.6. Actually, we notice that those two facts on hash functions are potentially useful in other scenarios, such as analyzing the security of digital signatures [13].

In addition, we notice that our analysis presented in this paper also applies to Wang et al.'s attack [12] against Fan et al.'s password authentication scheme [4]. In other words, Wang et al.'s attack is also *invalid* since they exploited the same attacking strategy as Yang et al. did in [15]; but Fan et al.'s scheme is also insecure because it is vulnerable to similar attacks as we identified in this paper. As the future work, we are considering to design password authentication schemes using smart cards with formal security.

References

1. M. Bellare and P. Rogaway, "Random oracles are practical: A paradigm for designing efficient protocols," *Proc. of the 1st ACM Conference on Computer and Communications Security* (*CCS'93*), pp. 62-73. ACM press, 1993.
2. D. Boneh and G. Durfee, "Cryptanalysis of RSA with private key d less than $N^{0.292}$," *IEEE Transactions on Information Theory*, vol. 46, no. 4, pp. 1339-1349, 2000.
3. C.K. Chan and L.M. Cheng, "Cryptanalysis of timestamp-based password authentication scheme," *Computers & Security*, vol. 21, no. 1, pp. 74-76, 2002.
4. L. Fan, J. H. Li, and H. W. Zhu, "An enhancement of timestamp-based password authentication scheme," *Computers & Security*, vol. 21, no. 7, pp. 665-667, 2002.

5. M.S. Hwang and L.H. Li, "A new remote user authentication scheme smart cards," *IEEE Transactions on Consumer Electronics*, vol. 46, pp. 28-30, 2000.
6. L. Lamport, "Password authentication with insecure communication," *Communications of the ACM*, vol. 24, pp. 770-772, 1981.
7. PKCS, "Public key cryptography standards, PKCS #1 v2.1," RSA Cryptography Standard, Draft 2, 2001. `http://www.rsasecurity.com/rsalabs/pkcs/`
8. 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.
9. J.J. Shen, C.W. Lin, and M.S. Hwang, "Security enhancement for the timestamp-based password authentication scheme using smart cards," *Computers & Security*, vol. 22, no. 7, pp. 591-595, 2003.
10. G. Tenenbaum. *Introduction to Analytic and Probabilistic Number Theory* (Theorem 5, page 41). Cambridge studies in advanced mathematics, Vol. 46. Cambridge University Press, 1995.
11. `http://mathworld.wolfram.com/RelativelyPrime.html`
12. B. Wang, J.-H. Li, and Z.-P. Tong, "Cryptanalysis of an enhanced timestamp-based password authentication scheme," *Computers & Security*, vol. 22, no. 7, pp. 643-645, 2003.
13. G. Wang, "On the security of a group signature scheme with forward security," *Proc. of the 6th Annual International Conference on Information Security and Cryptology* (*ICISC'03*), LNCS 2971, pp. 27-39. Springer-Verlag, 2004.
14. W.H. Yang and S.P. Shieh, "Password authentication schemes with smart cards," *Computers & Security*, vol. 18, no. 8, pp. 727-733, 1999.
15. C.-C. Yang, H.-W. Yang, and R.-C. Wang, "Cryptanalysis of security enhancement for the timestamp-based password authentication scheme using smart cards," *IEEE Transactions on Consumer Electronics*, vol. 50, No. 2, pp. 578-579, May 2004.
16. C.C. Yang, R.C. Wang, and T.Y. Chang, "An improvement of the Yang-Shieh password authentication schemes," *Applied Mathematics and Computation*, vol. 162, No. 3, pp. 1391-1396, 2005.
17. E.-J. Yoon, W.-H. Kim, and K.-Y. Yoo, "Security enhancement for password authentication schemes with smart cards," *Proc. of Trust and Privacy in Digital Business* (*TrustBus'05*), LNCS 3592, pp. 311-320. Springer-Verlag, 2005.