1-1-2014

A robust smart card-based anonymous user authentication protocol for wireless communications

Fengtong Wen
University of Wollongong, ftong@uow.edu.au

Willy Susilo
University of Wollongong, wsusilo@uow.edu.au

Guomin Yang
University of Wollongong, gyang@uow.edu.au

A robust smart card-based anonymous user authentication protocol for wireless communications

Abstract

Anonymous user authentication is an important but challenging task for wireless communications. In a recent paper, Das proposed a smart card-based anonymous user authentication protocol for wireless communications. The scheme can protect user privacy and is believed to be secure against a range of network attacks even if the secret information stored in the smart card is compromised. In this paper, we reanalyze the security of Das' scheme, and show that the scheme is in fact insecure against impersonation attacks. We then propose a new smart card-based anonymous user authentication protocol for wireless communications. Compared with the existing schemes, our protocol uses a different user authentication mechanism, which does not require different entities to maintain a synchronized clock.We show that the proposed new protocol can provide stronger security and better efficiency and scalability than previous schemes.

Keywords

remote user authentication, mobile computing, smart card, user anonymity, wireless network

Disciplines

Engineering | Science and Technology Studies

Publication Details

Wen, F., Susilo, W. & Yang, G. (2014). A robust smart card-based anonymous user authentication protocol for wireless communications. Security and Communication Networks, 7 (6), 987-993.

A Robust Smart Card Based Anonymous User Authentication Protocol for Wireless Communications

Abstract

Anonymous user authentication is an important but challenging task for wireless communications. In a recent paper, Das proposed a smart card based anonymous user authentication protocol for wireless communications. The scheme can protect user privacy and is believed to be secure against a range of network attacks even if the secret information stored in the smart card is compromised. In this paper, we reanalyze the security of Das' scheme, and show that the scheme is in fact insecure against impersonation attacks. We then propose a new smart card based anonymous user authentication protocol for wireless communications. Compared with the existing schemes, our protocol uses a different user authentication mechanism which does not require different entities to maintain a synchronized clock. We show that the proposed new protocol can provide stronger security and better efficiency and scalability than previous schemes.

Keywords: Remote user authentication; Mobile computing; Security; Smart card; User anonymity; Wireless network

1. Introduction

In recent years, with the fast development of mobile technologies, wireless networks have become widely available and interconnected. Remote user authentication is a fundamental research problem in wireless network security. When the mobile users (MU) roams into a foreign network, the Foreign Agent (FA) authenticates the roaming user with the help of the user's Home Agent (HA) [1, 7, 8].

As user privacy becomes a notable security issue in wireless communications, it is desirable to keep mobile users' identities anonymous in the remote user authentication process [12]. A lot of research work (e.g. [13, 6, 11, 2,

4, 9]) has been done in the design and analysis of anonymous user authentication protocols for wireless communications. However, most of the existing protocols were broken shortly after they were proposed, which shows the non-trivialness of the task.

In a very recent paper [3], Das showed the security weaknesses in some previous smart card based anonymous user authentication protocols. He then proposed a new scheme which is believed to be secure against a range of network attacks. However, in this paper, we show that there is also a serious security flaw in Das' anonymous user authentication scheme: an attacker who has stolen a user's smart card can successfully impersonate the user without knowing other secrets (such as the identity or password) of the user. Moreover, we show that Das' scheme cannot provide the strong replay resistance property which is claimed to be an advantage of his scheme over previous schemes.

We then propose a new smart card based anonymous user authentication protocol for wireless communications. Our protocol makes use of a user authentication mechanism which is different from the previous approaches and can successfully prevent different kinds of network attacks. In addition, the proposed scheme can achieve better efficiency and scalability compared with other smart card based anonymous user authentication protocols in the literature.

The paper is organized as follows. In Section 2, we briefly review Das's scheme. Then we show its weaknesses in Section 3. We then present our new protocol in Section 4 and analyze its security in Section 5. We compare the Performance of our new protocol with the previous schemes in Section 6. The paper is concluded in Section 7.

2. Review of Das's scheme

Let $G = Z_p$ denote a multiplication group and g a generator of G with order q, where p and q are large primes such that $p = 2q + 1$. The HA randomly selects a private key $S_{HA} = c \in Z_q$ and computes the public key $P_{HA} = g^c \ mod \ p$. Similarly, the foreign agent FA selects a private key $S_{FA} = e \in Z_q$ and computes the public key $P_{FA} = g^e \ mod \ p$. We summarize the notations used throughout the paper in Table 1.

Das' smart card based anonymous authentication scheme consists of four phases, namely registration phase, login phase, authentication phase, and password change phase. We only review the first three phases.

MU	The mobile user
ID_{MU}	Identity of MU
PW_{MU}	Password of MU
HA	Home agent of MU
ID_{HA}	Identity of HA
FA	Foreign agent of MU
ID_{FA}	Identity of FA
N	master secret key of HA
ctr_{MU}	A counter maintained by MU
$SK_{A,B}$	Session key shared between A and B
$H(\cdot)$	A secure collision-free one-way hash function
$E_K[M]$	Encrypting a message M using a symmetric key K
$D_K[C]$	Decrypting a ciphertext C using a symmetric key K
$Sign_{SK}[M]$	Signing a message M using a private key SK

Table 1: Notations

2.1. Registration phase

Step1. The MU chooses his/her own identity ID_{MU} and password PW_{MU}, and generates a large random number d. Then he/she computes the hash value $f = H(ID_{MU} \oplus PW_{MU} \oplus d)$, and sends the registration message ID_{MU}, f to the HA over a secure channel.

Step 2. Upon receiving the message from ID_{MU}, the HA computes $TK_{MU} = H(N \| ID_{MU}) \oplus f$ using its own secret master key N, and $r = ID_{HA} \oplus E_N[ID_{MU} \| m]$, where m is a random value which is different for each mobile user MU. The HA then issues a smart card to MU which contains $(TK_{MU}, H(\cdot), r)$ over a secure channel.

Step 3. Upon receiving the smart card, the MU securely stores d, f in the smart card. Hence, the smart card would contain $(TK_{MU}, H(\cdot), r, d, f)$.

2.2. Login phase

Step1. The MU first inserts his/her smart card into the device and then enters his/her identity ID_{MU} and password PW^*_{MU}. The smart card then computes $f^* = H(ID_{MU} \oplus PW^*_{MU} \oplus d)$ and verifies the equation $f^* = f$ using the stored value of f. If the equation holds, proceed to Step 2. Otherwise, the login phase is terminated immediately.

Step2. The smart card computes the value $TK^*_{MU} = TK_{MU} \oplus f = H(N \| ID_{MU})$. The smart card then acquires the current timestamp T_{MU} and generates a random nonce RN_{MU}. It further computes the followings: $A = g^a \ mod \ p, F = E_L[T_{MU}, RN_{MU}, ID_{FA}, A], M = E_{DH}[r]$, where $L = H(T_{MU} \oplus RN_{MU} \oplus TK^*_{MU})$ is a temporary key, a is a random number and $DH = P^a_{HA} \ mod \ p = g^{ac} \ mod \ p$ is the Diffie-Hellman key shared with the HA.

Step3. The smart card also computes $DH' = P^a_{FA} modp = g^{ea} \ mod \ p$ as the Diffie-Hellman key shared with the FA. The MU then computes the hash value $H(T_{MU} \oplus RN_{MU})$ and $U = E_{DH'}[M, F, ID_{HA}, T_{MU}, RN_{MU}]$, and sends the login request message $(A, H(T_{MU} \oplus RN_{MU}), U)$ to FA over a public channel.

2.3. Authentication phase

Step 1. After receiving the login request message from the MU, the FA computes $DH' = A^e \ mod \ p = g^{ea} \ mod \ p$ and decrypts U using DH' to reveal $M, F, ID_{HA}, T_{MU}, RN_{MU}$. The FA now knows that the MU's home agent is HA.

Step 2. The FA verifies the validity of T_{MU}. If it is valid, the FA computes $H(T_{MU} \oplus RN_{MU})$ and compares this value with the received value to verify the validity of RN_{MU}. In order to protect replay attacks,The FA stores (ID_{MU}, RN_{MU}) in its database. When the FA receives another login message, say $[A', H(T'_{MU} \oplus RN'_{MU}), U']$, it first retrieves random nonce RN'_{MU} by decrypting U'. If the timestamp T'_{MU} and the random nonce RN'_{MU} are valid, the FA matches the retrieved RN'_{MU} with the stored random nonce RN_{MU} corresponding to MU in its database. If the the two random numbers match, the FA rejects this login request message. Otherwise, the FA replaces the old random nonce RN_{MU} by the new one RN'_{MU} for the user MU in its database.

Step 3. The FA computes $B = g^b \ mod \ p$ and $DH'' = P^b_{HA} \ mod \ p = g^{cb} \ mod \ p$ as the key shared with the HA, where $b \in Z_q$ is a secret random number generated by the FA. The FA acquires the current timestamp T_{FA} and generates a random nonce RN_{FA}. The FA then computes a signature $V = Sign_{S_{FA}}[H(A \| B \| M \| F \| T_{MU} \| RN_{MU} \| T_{FA} \| RN_{FA} \| Cert_{FA})]$ where $Cert_{FA}$ is FA's certificate and S_{FA} is its private key, $W = E_{DH''}[A, B, M, F, T_{MU}, RN_{MU}, T_{FA}, RN_{FA}, V, Cert_{FA}]$ and $H(T_{FA} \oplus RN_{FA})$. FA then sends the message $(B, H(T_{FA} \oplus RN_{FA}), W)$ to the HA.

Step 4. HA computes $D_{DH''}[W] = (A, B, M, F, T_{MU}, RN_{MU}, T_{FA}, RN_{FA}, V, Cert_{FA})$, where $DH'' = B^c \ mod \ p = g^{cb} \ mod \ p$. The HA verifies the validity of T_{FA}, RN_{FA}, $Cert_{FA}$ and FA's signature V in turn. If these are valid, the HA computes $ID_{HA} \oplus D_{DH}[M] = E_N[ID_{MU} \| m]$ to reveal ID_{MU}. By searching ID_{MU} in its database, the HA verify whether the MU is a legal user or not.

Step 5. The HA computes $L = H(T_{MU} \oplus RN_{MU} \oplus H(N \| ID_{MU}))$ and decrypts F using L to reveal $T_{MU}, RN_{MU}, ID_{FA}, A$. The HA checks whether the decrypted T_{MU}, RN_{MU} and ID_{FA} are correct. If any value is invalid, the HA notifies the FA that MU is not a legal user. Otherwise, the HA goes to the next Step. In addition, the HA adopted the same method as in Step 2 of the Authentication Phase to prevent replay attacks.

Step 6. The HA computes $D = g^x \ mod \ p$ and the session secret key shared with the FA as $SK_{FA,HA} = B^x \ mod \ p = g^{bx} \ mod \ p$, where $x \in Z_q$ is a secret random number. Then, HA computes $X = Sign_{S_{HA}}[H(A \| B \| D \| T_{HA} \| RN_{MU} \| RN_{FA} \| Cert_{HA})]$, where S_{HA} is the private key of HA, and $Y = E_{SK_{FA,HA}}[H(H(N \| ID_{MU}) \| D) \| A \| B \| D \| T_{HA} \| RN_{MU} \| RN_{FA} \| X \| Cert_{HA}]$. HA computes the secret session key shared with MU as $SK_{MU,HA} = A^x \ mod \ p = g^{ax} \ mod \ p$. HA then sends the message $(D, H(T_{HA} \oplus RN_{MU} \oplus RN_{FA}), Y)$ to FA.

Step 7. FA computes $SK_{FA,HA} = D^b \ mod \ p = g^{bx} \ mod \ p$ and decrypts Y to retrieve the information $H(H(N \| ID_{MU}) \| D), A, B, D, T_{HA}, RN_{MU}, RN_{FA}, X, Cert_{HA}$. The FA then verifies the validity of T_{HA}, RN_{MU}, RN_{FA} by using the received $H(T_{HA} \oplus RN_{MU} \oplus RN_{FA})$. If these are valid, FA verifies HA's certificate $Cert_{HA}$ and signature X. If they are also valid, FA proceeds to the next Step.

Step 8. FA computes $SK_{FA,MU} = A^b \ mod \ p = g^{ab} \ mod \ p$ and $Z = E_{SK_{FA,MU}}[TCert_{MU} \| H(H(N \| ID_{MU}) \| D) \| A \| B \| D \| RN_{MU}]$ where $TCert_{MU}$ is a temporary certificate for MU, and sends $(B, H(RN_{MU}), Z)$ to MU.

Step 9. MU computes the secret session key shared with FA as $SK_{MU,FA} = B^a \ mod \ p = g^{ab} \ mod \ p$ and decrypts Z to retrieve $TCert_{MU}, H(H(N \| ID_{MU}) \| D), A, B, D, RN_{MU}$. MU computes $H(RN_{MU})$ using RN_{MU} generated in the login phase and compares it with the received hash value $H(RN_{MU})$. If they match, MU verifies the certificate $TCert_{MU}$ using FA's public key P_{FA}. If it is valid, MU computes $H(H(N \| ID_{MU}) \| D)$ using its computed $TK^*_{MU} = H(N \| ID_{MU})$ in login phase and the retrieved value of D. If it matches with

the decrypted $H(H(N\|ID_{MU})\|D)$, MU confirms that FA is verified by HA. Then MU computes the session key $SK_{MU,HA} = D^a \ mod \ p = g^{ax} \ mod \ p$ shared with the HA for future secure communication.

3. Security Analysis of Das's scheme

3.1. Security against Impersonation

In this section, we show that Das's scheme is vulnerable to impersonation attacks if a user's smart card is stolen and an attacker has extracted all the information $(TK_{MU}, H(\cdot), r, d, f)$ stored in the smart card (e.g. by power analysis [5, 10]). The attacker then impersonate the user MU as follows.

In login phase, the attacker acquires the current timestamp T^* and generates two random number RN^*, a^*. He/she then computes the following values $TK^*_{MU} = TK_{MU} \oplus f = H(N\|ID_{MU})$, $A^* = g^{a^*} \ mod \ p$, $L^* = H(T^* \oplus RN^* \oplus TK^*_{MU})$, $F^* = E_{L^*}[T^*, RN^*, ID_{FA}, A^*]$, $DH^* = P^{a^*}_{HA} = g^{a^*c} \ mod \ p$, $M^* = E_{DH^*}[r]$, $DH^{*'} = P^{a^*}_{FA} = g^{a^*e} \ mod \ p$, $U^* = E_{DH^{*'}}[M^*, F^*, ID_{HA}, T^*, RN^*]$, and sends the login request message $(A^*, H(T^* \oplus RN^*), U^*)$ to FA.

It is easy to see that the login request is in the correct format. Upon receiving the request, FA and HA will execute the protocol normally and the attacker will pass the authentication successfully. In the above attack, the attacker needs to know the identity of the home agent HA. However, due to the small number of home agents (compared with the number of mobile users), the adversary can perform the attack simply by a brute-force search on the possible home agents.

3.2. Security against Strong Replay Attack

In [3], Das claimed that timestamps are not an ideal solution to prevent replay attacks since replay attacks may still occur within the time period specified by the timestamps. He then proposed a new approach to avoid replay attacks: FA and HA will store (ID_{MU}, RN_{MU}) and $\{(ID_{MU}, ID_{FA}), (RN_{MU}, RN_{FA})\}$ respectively in their databases. Later, when FA (or HA) receives a new request, it will check if the random number in this new request is the same as the one in its database. If they are the same, the request is identified as a replay attack and will be rejected. Otherwise, the normal procedures will be performed and the new random number will replace the old one in the database.

First of all, there is a mistake in Das' scheme: such a procedure to prevent replay attacks cannot be performed by FA since throughout the whole protocol the identity ID_{MU} of a mobile user will never be revealed to FA. Secondly, we believe such a strategy is still not strong enough to prevent replay attack when considering the following attacking scenario: suppose MU executes two sessions within a very short time interval such that the timestamp in the first session is still valid when the second session has been completed. Since the databases of FA and HA have been updated, the attacker can now replay the first request in order to pass the authentication.

4. A New Smart Cased based Anonymous Authentication Protocol

We propose a new smart card based anonymous authentication protocol for wireless communications. Notably, the new protocol makes use of a counter-based authentication mechanism to provide strong replay resistance. The new protocol can also resist a range of attacks such as offline password guessing attacks and impersonation attacks even if the smart card is stolen.

The system parameters p, q, g, G are the same as in Das' scheme. Let $(P_{HA} = g^c, S_{HA} = c)$ and $(P_{FA} = g^e, S_{FA} = e)$ denote the public and private keys of HA and FA respectively. The new protocol has four phases: registration, login, authentication, and password change.

4.1. Registration phase

1). The MU chooses his/her own identity ID_{MU} and password PW_{MU}, and generates a large random number d. He/she then computes the hash value $f = H(ID_{MU} \| PW_{MU} \| d)$, and sends the registration message ID_{MU}, f to the HA via a secure channel.

2). Upon receiving the request, HA computes $TK_{MU} = H(N \| ID_{MU}) \oplus f$ using its own master secret key N, and $r = ID_{HA} \oplus E_N[ID_{MU} \| m]$, where m is a random value which is different for each mobile user. HA then initiates a counter $ctr_{MU} = 0$ for MU and creates a record (ID_{MU}, ctr_{MU}) in its database. HA then issues a smart card to MU which contains $(TK_{MU}, H(\cdot), r,$ $ctr_{MU})$ over a secure channel.

3). Upon receiving the smart card, MU computes the hash value $f^* = H(ID_{MU} \oplus PW_{MU} \oplus d)$, and stores f^*, d in the smart card. So the smart card contains $(TK_{MU}, H(\cdot), r, ctr_{MU}, f^*, d)$.

4.2. Login phase

1). MU first inserts his/her smart card into the device and then enters his/her identity ID_{MU} and password PW^*_{MU}. The smart card then computes $f^{*'} = H(ID_{MU} \oplus PW^*_{MU} \oplus d)$ and verifies the condition $f^{*'} = f^*$ using the stored value of f^*. If the equation holds, the smart card ensures that MU enters his/her identity and password correctly and proceed to the next step. Otherwise, the login phase is terminated immediately.

2). The smart card computes $TK^*_{MU} = TK_{MU} \oplus H(ID_{MU} \| PW_{MU} \| d) = H(N \| ID_{MU})$, $ctr_{MU} = ctr_{MU} + 1$, $A = g^a \ mod \ p, F = E_L[ctr_{MU}, ID_{FA}, A]$, $M = E_{DH}[r]$, where $L = H(ctr_{MU} \oplus TK^*_{MU})$ is a temporary key, $a \in Z_q$ is a random number and $DH = P^a_{HA} = g^{ac} \ mod \ p$ is a Diffie-Hellman key shared with HA.

3). The smart card also computes $DH' = P^a_{FA} = g^{ea} \ mod \ p$ as the Diffie-Hellman key shared with FA, and $U = E_{DH'}[M, F, ID_{HA}, ctr_{MU}]$, and sends the login request message (A, U) to FA over a public channel.

4.3. Authentication phase

Upon receiving the login request message (A, U) from the MU, the FA and the HA perform the following steps.

1). FA computes $DH' = A^e = g^{ea} \ mod \ p$ and decrypts U using DH' to recover M, F, ID_{HA}, ctr_{MU}. Then FA knows that MU's home agent is HA.

2). $FA \rightarrow HA : (B, W)$. FA computes $B = g^b \ mod \ p$ and $DH'' = P^b_{HA} = g^{cb} \ mod \ p$ as the key shared with the HA, where $b \in Z_q$ is a random number generated by the FA. FA then computes $V = Sign_{S_{FA}}[H(A \| B \| M \| F \| Cert_{FA} \| ctr_{MU})]$ and $W = E_{DH''}[A, B, M, F, V, Cert_{FA}, ctr_{MU}]$, where S_{FA} is the private key of FA and $Cert_{FA}$ is its certificate. Then FA sends the message (B, W) to HA.

3). Upon receiving the message (B, W) from FA, HA computes $D_{DH''}[W] = (A, B, M, F, V, Cert_{FA}, ctr_{MU})$, where $DH'' = B^c = g^{cb} \ mod \ p$. The HA verifies $Cert_{FA}$ and FA's signature V using FA's public key P_{FA} included in $Cert_{FA}$. If they are valid, the HA computes $D_N[ID_{HA} \oplus D_{DH}[M]] = ID_{MU} \| m$ to recover ID_{MU}. By matching ID_{MU} in its database, the HA verify whether the MU is a legal user or not. If ID_{MU} doesn't exist then HA notifies FA that MU is not a legal user.

4). HA computes $L = H(ctr_{MU} \oplus H(N \| ID_{MU}))$ and decrypts F using L to recover (ctr_{MU}, ID_{FA}, A). HA checks whether the decrypted ctr_{MU}, A and ID_{FA} are same as the received ctr_{MU}, A and the identity of FA in $Cert_{FA}$. If the verification fails, HA notifies FA that the user authentication fails.

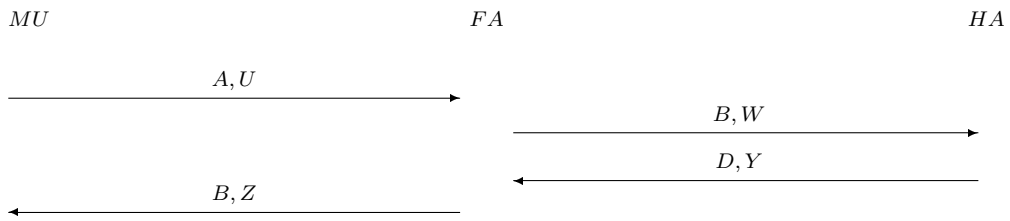

Figure 1: Message Flows in Authentication Phase.

In order to prevent replay attacks, the HA also verifies the retrieved ctr_{MU} with the stored counter ctr'_{MU} corresponding to ID_{MU}. If $ctr_{MU} > ctr'_{MU}$ then the HA replaces ctr'_{MU} with ctr_{MU} in its database and proceeds to the next step. Otherwise, HA notifies FA that the user authentication fails.

5). $HA \rightarrow FA : (D, Y)$. The HA computes $D = g^x \ mod \ p$ and the session secret key shared with the FA as $SK_{FA,HA} = B^x = g^{bx} \ mod \ p$ where $x \in Z_q$ is a random number generated by HA. HA then computes $X = Sign_{S_{HA}}[H(A\|B\|D\|Cert_{HA})]$ and $Y = E_{SK_{FA,HA}}[H(H(N\|ID_{MU})\|A \|B\|D)\|A\|B\|D\|X\|Cert_{HA}]$ where S_{HA} is the private key of the HA and $Cert_{HA}$ is its certificate. HA also computes the secret session key shared with MU as $SK_{MU,HA} = A^x = g^{ax} \ mod \ p$. HA then sends the message (D, Y) to FA.

6). Upon receiving the message (D, Y) from HA. FA computes $SK_{FA,HA} = D^b = g^{bx} \ mod \ p$ and decrypts Y to recover the values $H(H(N\|ID_{MU})\|A\|B\|D)$, $A, B, D, X, Cert_{HA}$. FA verifies HA's certificate $Cert_{HA}$ and signature X. If they are valid, FA proceeds to the next Step.

7). $FA \rightarrow MU : (B, Z)$. FA computes $SK_{FA,MU} = A^b = g^{ab} \ mod \ p$ and $Z = E_{SK_{FA,MU}}[Cert_{FA}\|TCert_{MU}\|H(H(N\|ID_{MU})\|A\|B\|D)\|A\|B\|D]$, where $TCert_{MU}$ is a temporary certificate for MU. FA then sends (B, Z) to the MU.

8). MU computes the secret session key shared with the FA as $SK_{MU,FA} = B^a = g^{ab} \ mod \ p$ and uses it to decrypt Z to recover $TCert_{MU}, H(H(N\|ID_{MU})$ $\|A\|B\|D), A, B, D$. The MU then verifies $Cert_{FA}$ and the temporary certificate $TCert_{MU}$ using the public key P_{FA} of FA. If the verification is successful, MU computes $H(TK^*_{MU}\|A\|B\|D)$ and compares it with the decrypted $H(H(N\|ID_{MU})\|A\|B\|D)$. If the two values are equal, MU accepts the session and computes $SK_{MU,HA} = D^a = g^{ax} \ mod \ p$ as the session key shared with HA.

4.4. Password change phase

1) MU inserts his/her smart card into the device and enters his/her identity ID_{MU} and old password PW_{MU}^{old}.

2) The smart card computes $f^{*'} = H(ID_{MU} \oplus PW_{MU}^{old} \oplus d)$ and verifies the condition $f^{*'} = f^*$ using the stored f^*. If the equation does not hold, the smart card rejects the password change request. Otherwise, MU enters his/her new password PW_{MU}^{new}. The smart card computes $f^{*''} = H(ID_{MU} \oplus PW_{MU}^{new} \oplus d)$ and $TK_{MU}^{new} = TK_{MU}^{old} \oplus H(ID_{MU} \| PW_{MU}^{old} \| d) \oplus H(ID_{MU} \| PW_{MU}^{new} \| d)$, and replaces the old values of (TK_{MU}^{old}, f^*) with $(TK_{MU}^{new}, f^{*''})$ respectively.

5. Security Analysis of The Proposed Scheme

In this section, we show that our proposed scheme can resist different types of attacks and provide user anonymity.

5.1. User Anonymity

From Figure 1, we can see that the communication transcript reveals no information about the identity ID_{MU} of the user. First of all, $A = g^a, B = g^b, D = g^x$ are all random numbers which are not related to ID_{MU}. Secondly, U, W, Y and Z are all ciphertexts encrypted under different Diffie-Hellman keys. Because of the intractability of the Diffie-Hellman problem, the adversary is unable to decrypt U, W, Y or Z to recover the encrypted message.

From the view point of the foreign agent FA, it can decrypt U to recover $(M, F, ID_{HA}, ctr_{MU})$ and decrypt Y to recover $(H(H(N \| ID_{MU}) \| A \| B \| D)$, $A, B, D,\ X, Cert_{HA})$. Since M and F are ciphertexts which can only be decrypted by HA, FA will not learn any information about ID_{MU} from these values. Also, since the hash function H is one-way, FA cannot learn the identity of MU from $H(H(N \| ID_{MU}) A \| B \| \| D)$. Finally, $X = Sign_{S_{HA}}[H(A \| B \| D \| Cert_{HA})]$ also reveals no information about ID_{MU} to FA. Therefore, the foreign agent is not able to recover the identity of MU.

5.2. Security against Impersonation

We consider three types of impersonation attacks, namely MU impersonation attack, FA impersonation attack, and HA impersonation attack.

MU Impersonation. In order to impersonate the MU, the adversary must obtain the value of $H(N \| ID_{MU})$. We then consider two situations: (1) the

adversary compromises the user identity and password but the smart card is secure; and (2) the adversary compromises the smart card but the the user identity and password are secure.

In the first case, it is obvious that adversary is unable to obtain the value of $H(N\|ID_{MU})$ since the smart card is secure. Notice that $H(N\|ID_{MU})$ is concealed in $TK_{MU} = H(N\|ID_{MU}) \oplus H(ID_{MU}\|PW_{MU}\|d)$ in the smart card. Since the smart card is secure, the adversary is unable to obtain TK_{MU} or $H(N\|ID_{MU})$.

In the second case, when the smart card is stolen and compromised, the adversary can learn the values of $(TK_{MU}, H(\cdot), r, ctr_{MU}, f^*, d)$ in the smart card. However, this time the adversary knows neither ID_{MU} nor PW_{MU}, and again she cannot compute the value of $H(N\|ID_{MU})$. We must also consider **offline password guessing attacks** in this case, that is the adversary uses a brute force search to find out the correct password. In the proposed anonymous authentication scheme, the user identity is protected against outsiders and the FA, this can help prevent the password guessing attack. Since there can be a huge number of users in the mobile system, it is infeasible for an adversary to do an exhaustive search for all the possible (ID, password) pairs. However, we remark that if the user ID space is small, then offline password guessing attack would become feasible if the smart card of a user is stolen and compromised by the attacker.

It is worth noting that FA can also learn the value of $(H(H(N\|ID_{MU})\|A\|B\|D)$ in the authentication phase. However, due to the one-wayness of the hash function, FA is unable to obtain the value of $H(N\|ID_{MU})$.

FA/HA Impersonation. Since FA and HA uses digital signatures to authenticate each other, the adversary is unable to impersonate either of them due to the unforgeability of the digital signature scheme. Also, it is infeasible for FA to impersonate HA to the mobile user since the mobile user will verify the message $(H(H(N\|ID_{MU})\|A\|B\|D)$ in order to make sure that the value of D is indeed generated by the HA.

5.3. Strong Replay Resistance

We have used the counter based authentication mechanism to prevent replay attacks. The advantage of using this mechanism compared with the timestamp approach is that we don't require different parties in the system to maintain a synchronized clock. It also can prevent the replay attack presented in Section 3.2.

It is obvious that if the adversary replays the first or second message, then HA will detect the attack when examining the counter ctr_{MU} of the user MU. For third and fourth messages, we used the challenge-response mechanism to prevent replay attack. In the third message, the signature X is generated by HA based on the challenge B sent by FA, and in the fourth message, the authentication token $(H(H(N\|ID_{MU})\|A\|B\|D)$ is generated based on the challenge A sent by MU. Therefore, the adversary cannot replay any message in the protocol.

5.4. Secure Key Establishment and Forward Secrecy

At the end of the protocol, the MU will establish a session key $SK_{MU,FA} = B^a = g^{ab} \ mod \ p$ with FA, and another session key $SK_{MU,HA} = D^a = g^{ax} \ mod \ p$ with HA. Since all the messages are protected against impersonation and replay attacks, the MU can guarantee that the session keys are established with the intended peer entities. Due to the intractability of the Diffie-Hellman problem, the adversary is unable to derive any session key from the values of A, B, D.

It is also worth noting that the proposed protocol achieves perfect forward secrecy. Since the values of A, B, D are freshly generated in each session, all the past session keys will remain secure even if the long-terms secret keys are compromised at a later stage.

6. Performance Comparison

We compare our new scheme with two recently proposed smart card based anonymous user authentication schemes due to Das [3] and Li et al. [9]. In Table 2, we provide the comparison based on the key security and usability features of these schemes, while in Table 3, we compare their efficiency.

From the tables, we can see that our proposed scheme provides better security and usability than the other two schemes. In particular, one special feature of our scheme is that we don't require all the parties (namely MU, FA, and HA) in the system to maintain a synchronized clock. This is important in the wireless roaming scenario since clients and servers in different domains may use different clocks. Therefore, our scheme is more scalable than the previous schemes.

Feature	Li et al. [9]	Das [3]	Ours
Correct Password Update	No	Yes	Yes
Server Knows No Password	Yes	Yes	Yes
No Synchronized Clock	No	No	Yes
User Anonymity	Yes	Yes	Yes
Mutual Authentication	Yes	No	Yes
Strong Replay Resistance	No	No	Yes
Key Agreement	Yes	Yes	Yes
Forward Secrecy	Yes	Yes	Yes

Table 2: Security and Usability Comparison

	Li et al. [9]	Das [3]	Ours
Modular Exponentiation	6+6pre*	6+6pre	6+6pre
Hash Operation	10	15	10
Symmetric-key Encryption	6	6	6
Symmetric-key Decryption	7	7	7
Signature Generation	2	2	2
Signature Verification	2	2	2

* pre denotes pre-computation

Table 3: Efficiency Comparison

7. Conclusion

In this paper, we showed some security flaws in a recently proposed smart card based anonymous user authentication scheme. We also proposed a new scheme which can achieve stronger security and better usability than the existing schemes without sacrificing efficiency. Our scheme does not require entities in different domains to maintain a synchronized clock, and hence is more suitable for wireless roaming networks.

References

[1] C. C. Chang, J. S. Lee, and Y. F. Chang, Efficient authentication protocols of GSM, Comput. Commun. 28(8):921-928, 2005.

[2] C. C. Chang, C. Y. Lee, and Y. C. Chiu, Enhanced authentication scheme with anonymity for roaming service in global mobility networks, Comput. Commun. 32(4):611-618, 2009.

[3] A. K. Das, A secure and effective user authentication and privacy preserving protocol with smart cards for wireless communications, Networking Science, doi:10.1007/S13119-012-0009-8, 2012.

[4] D. He, M. Ma, Y. Zhang, C. Chen, and J. Bu, A strong user authentication scheme with smart cards for wireless communications, Comput. Commun. 34(3):367-374, 2011.

[5] P. Kocher, J. Jaffe, and B. Jun, Differential power analysis, Advances in Cryptology - CRYPTO, LNCS 1666, pp.388-397, 1999.

[6] C. C. Lee, M. S. Hwang, and I. E. Liao, Security enhancement on a new authentication scheme with anonymity for wireless environments, IEEE Trans. Ind. Electron. 53(5):1683-1686, 2006.

[7] C .C. Lee, M. S. Hwang, and I. E. Liao, A new authentication protocol based on pointer forwarding for mobile communications, Wireless Commun. Mobile Comput. 8(5):661-672, 2008.

[8] C. C. Lee, M. S. Hwang, and W. P. Yang, Extension of authentication protocol for GSM, IEE Proc. Commun. 150(2):91-95, 2003.

[9] C. T. Li and C. C. Lee, A novel user authentication and privacy preserving scheme with smart cards for wireless communications, Math. Comput. Model. 55(1-2):35-44, 2012.

[10] T. S. Messerges, E. A. Dabbish, and R. H. Sloan. Examining smart-card security under the threat of power analysis attacks, IEEE Trans Comput 5(51): 541-552, 2002.

[11] C. C. Wu, W. B. Lee, and W. J. Tsaur, A secure authentication scheme with anonymity for wireless communications, IEEE Commun. Lett. 12(10):722-723, 2008.

[12] G. Yang, D. S. Wong, X. Deng, Anonymous and Authenticated Key Exchange for Roaming Networks, IEEE Trans. Wireless Commun. 6(9): 3461-3472, 2007.

[13] J. Zhu and J. Ma, A new authentication scheme with anonymity for wireless environments, IEEE Trans. Consum. Electron. 51(1):230-234, 2004.