A New Secure Anonymous Protocol for Distributed Computer Networks

Cui Hui[1], Cao Tianjie[1,2]
[1]School of Computer, China University of Mining and Technology
Xuzhou, Jiangsu, 221116, China
snager@163.com
[2]National Mobile Communications Research Laboratory, Southeast University
Nanjing, Jiangsu, 210096, China
tjcao@cumt.edu.cn

***Abstract*—In this paper, we analyzed Hsu-Chuang scheme and presented the vulnerability. We further secured their protocol by proposing a novel protocol that overcomes the above limitation while achieving the same security features. Besides, we made a demonstration of the correctness of our new scheme and analyzed the security properties. After compared Hsu-Chuang scheme and the new scheme, we also illustrated an example to illustrate the application of our improved protocol.**

***Index Terms*—key agreement, user anonymity, mutual authentication, forward security, backward security.**

I. INTRODUCTION

In the distributed computing environments, entity authentication is crucial to prevent unauthorized entities from obtaining system resources. Usually, the process of authentication involves the exchanges of entities' identities, and authenticated key generation, etc. However, in some applications, it is desirable or even necessary to maintain entity anonymity, because a lot of sensitive information such as entity's identification, entity movement, individual preferences or web surfing patterns can be collected if the entities identifications are not protected during the authentication process. Therefore, those schemes that do not preserve users' anonymity cannot apply well to those environments where the protection of entities' identities is required.

In 2000, Lee and Chang [1] proposed a user identification scheme with key distribution maintaining user anonymity for distributed computer networks. Wu and Hsu, however, showed that the Lee-Chang scheme is insecure against impersonation and identity disclosure attacks [2]. An adversary can plot an impersonation attack to masquerade as a service provider in order to exchange a session key with a user without being detected in the authentication protocol. In addition, an adversary can plot the disclosure attack to identify a user who requests services with a released session key. Wu and Hsu further proposed an improved scheme to withstand these two attacks, preserving the same security requirements as those of the Lee-Chang scheme.

Recently, Yang et al. demonstrated a compromising attack whereby it is possible for an adversary to derive the private keys of users who request services [3]. Yang et al. also proposed an improvement to the Wu-Hsu scheme to eliminate such a security leak and achieve the same security requirements. Mangipudi and Katti [4] later demonstrated a Deniable-of-Service (DoS) attack on the Yang et al. scheme. An adversary can cause the server to deny the requests of a legitimate user by tampering with the transmitted messages. In order to withstand such a DoS attack, Mangipudi and Katti further employed a digital signature to protect the integrity of the transmitted messages.

Hsu and Chuang [5], however, demonstrated an identity disclosure attack on the Yang et al. and Mangipudi-Katti schemes [4] to show that their claimed user anonymity requirement is violated. That is, the identity of the communicating user can be easily ascertained from the exchanged messages. They proposed a novel user identification scheme with key distribution preserving user anonymity which eliminates these security leaks and achieves all of the above-mentioned properties.

In this paper, we proposed a novel user identification scheme with key distribution preserving user anonymity (SAIKA) which eliminated the security vulnerability well and achieves all the security properties of previous schemes.

We intend to design a secure key agreement protocol for distributed computer networks, which is expected to inherit all the good virtues of the previous schemes and some added security properties. Here we summarize all these requirements to evaluate our new scheme as follows.

User Anonymity: The scheme should preserve the

user's identity, namely, a server could not tell a user's identity. Once the connection between the user and the server has been established, the probability of the server to guess the user's identity is 1/*n*, where *n* is the number of ring members.

Security of Session Key: The scheme should preserve the security of session key, that is to say, when executing our improved protocol, except the correspondence patties nobody outside could acquire the session key.

Mutual Authentication: The scheme should assure that not only can the server verify the legal user, but the user can also verify the server. In authenticated protocols, mutual authentication is an important attribute, so our scheme should also be in favor of it perfectly.

Forward Secrecy and Backward Secrecy: The scheme should satisfy forward secrecy and backward secrecy, namely, if the session key generated in *j* period has been leaked, the attacker can't forge any session key generated before *j* period or after *j* period. Therefore, the scheme should defeat some attacks like replay attack and so on.

II. THE PROPOSED SCHEME

Similar to Hsu-Chuang scheme [5], our proposed scheme (SAIKA) also consists of three types of entities (a SCPC, the users and the servers). We start by present our scheme in detail, and then discuss the security of the concrete scheme.

A. Protocol description

The SCPC chooses two large primes p and q, computes $N = pq$, and determines (e, d) such that $ed = 1 \bmod \phi(N)$, where $\phi(N) = (p - 1)(q - 1)$. The SCPC randomly chooses an element $g \in Z_N^*$, which is the generator of both Z_p^* and Z_q^*. The SCPC determines a symmetric-key cryptosystem. The symbol $E_K(m)$ denotes the encryption of the message m with a key K, and the symbol $D_K(c)$ denotes the decryption of the cipher text c with a key K. The SCPC determines a collision free secure one-way hash function $h(.)$. Finally, the SCPC publishes (e, N, g) as public system parameters, while maintaining the secrecy of (d, p, q).

(1) Registration phase

The user U_i collects a group of n identities, and composes an identity list $L = \{ID_1, \ldots, ID_n\}$, including U_i's own identity $ID_i = ID_s$ ($s \in [1, n]$). Then the user U_i submits L to the SCPC for registration. On receiving L, the SCPC uses a private key d to generate U_i's private key $\{S_i\}$ as $S_i = ID_i^d \bmod N$ ($i \in [1, n]$). Then, the SCPC sends the private key $\{S_i\}$ to U_i via a secure channel.

The service provider P_j submits an ID_j to the SCPC for registration. On receiving ID_j, the SCPC uses a private key d to generate P_j's private key S_j as $S_j = ID_j^d \bmod N$. Then, the SCPC sends the private key S_j to or P_j via a secure channel.

(2) Anonymous authentication and key agreement phase

If user U_i wants to gain access privilege from service provider P_j, U_i and P_j will cooperatively perform the following steps.

1. The user U_i submits the service request to the service provider P_j. U_i chooses a random number r, computes

$$C = r \oplus h(ID_j \| T_0),$$

and then sends (C, T_0) to P_j.

2. P_j chooses a random number k, computes

$$r = C \oplus h(ID_j \| T_0)$$
$$Z = g^k S_j^r \bmod N,$$

and then sends Z back to U_i.

3. After receiving Z from P_j, U_i chooses a random number t and computes:

$$\alpha = Z^e ID_j^{-r} \bmod N,$$
$$K_{ij} = \alpha^t \bmod N,$$
$$w = g^{et} \bmod N.$$

4. For every $1 \le i \le n$, P_j computes

$$x_i = g^t S_i^{h(K_{ij} \| Z \| w \| T)} \bmod N,$$
$$y = E_{K_{ij}}(ID_1, \ldots, ID_n)$$

where T is the current timestamp. Note that K_{ij} is regarded as the session key shared between U_i and P_j. Then, U_i sends the message $(w, \{x_i\}, y, T)$ to P_j.

5. When P_j receives $(w, \{x_i\}, y, T)$, the validity of T is first checked. If it is invalid, P_j aborts the protocol; otherwise, K_{ij} is derived via the following equation:

$$K_{ij} = w^k \bmod N.$$

6. P_j uses the key K_{ij} to decrypt y as

$$L = D_{K_{ij}}(y),$$

and verifies the validity of the recovered identifier ID_i by checking

$$w ID_i^{h(K_{ij} \| Z \| w \| T)} \stackrel{?}{=} x_i^e \pmod N.$$

If this equation holds, P_j is convinced that U_i is an authorized user.

7. P_j computes D_i as

$$D_i = h(K_{ij} \| T' \| \{ID_i\} \| ID_j)$$

and sends $(D_i, T')$to U_i, where T' is the current timestamp.

8. Upon receiving (D_i, T') from P_j, U_i first checks the validity of T'. If it is a valid timestamp, U_i computes

$D_i' = h(K_{ij}||T'||\{ID_i\}||ID_j)$ and checks if D_i' is identical to the received D_i. If it holds, U_i is convinced that P_j is the valid service provider.

(3) Reveal User's Identity

If P_j wants to confirm that U_i is the real user, he could send a request to SCPC, and SCPC will verify U_i's identity.

1. P_j computes $E_{S_j}(\{x_i\}||T_1)$, and sends it to SCPC as the request to verify U_i's identity.

2. U_i computes x_s as

$$x_s = g^t S_s^{h(K_{ij}||Z||w||T)} \bmod N \quad \text{and} \quad E_{S_i}(x_s||T_2),$$

and then sends $E_{S_i}(x_s||T_2)$ to SCPC.

3. SCPC encrypts $E_{S_j}(\{x_i\}||T_1)$ and $E_{S_i}(x_s||T_2)$, and then verifies x_s. if there is x_i ($\in \{x_i\}$) is equal to x_s, SCPC believes that U_i is the real user, and sends $\{x_i, T_2\}$ to P_j.

B. Correctness proof

In this section we will give the correctness demonstration of our scheme.

(1) $K_{ij} \overset{?}{=} K_{ij}$

$K_{ij} = \alpha^t \bmod N$
$= (Z^e ID_j^{-1})^t \bmod N$
$= ((g^k S_j)^e ID_j^{-1})^t \bmod N$
$= (g^{etk}) \bmod N$
$= (g^{et})^k \bmod N$
$= w^k \bmod N$

(2) $wID_i^{h(K_{ij}||Z||w||T)} \overset{?}{=} x_i^e (\bmod N)$

$$wID_i^{h(K_{ij}||Z||w||T)} = g^{et} ID_i^{h(K_{ij}||Z||w||T)} (\bmod N)$$
$$= g^{et} ID_i^{h(K_{ij}||Z||w||T)ed} (\bmod N)$$
$$= (g^t ID_i^{dh(K_{ij}||Z||w||T)})^e (\bmod N)$$
$$= (g^t S_i^{h(K_{ij}||Z||w||T)})^e (\bmod N)$$
$$= x_i^e (\bmod N)$$

C. Security analysis

Here we will discuss the security properties of our proposed scheme, so as to show our scheme meets all the requirements presented in section 1.

User Anonymity: After receiving Z from P_j, U_i chooses a random number t, and then for every $1 \le i \le n$, P_j computes $x_i = g^t S_i^{h(K_{ij}||Z||w||T)} \bmod N$ to compose $\{x_i\}$. Besides it, P_j could use the key K_{ij} to decrypt y and get $L = \{ID_1, \ldots, ID_n\}$. But P_j can not tell the identity of the user from all these information, and the only clue P_j can get is that the user is someone among the list $L = \{ID_1, \ldots, ID_n\}$.

It should be pointed out that P_j in our scheme is always assumed to be honest. Indeed, a malicious service provider could learn whether the identity of user is the guessed identity $ID_i \in L$ for some i by choosing a unique ID^* and some random numbers for computing $x_i = g^t S_i^{h(K_{ij}||Z||w||T)} \bmod N$. If the equation can pass the verification $wID_i^{h(K_{ij}||Z||w||T)} = x_i^e (\bmod N)$, then P_j could learn that the identity of user is ID^*. However, such a cheating could be successful with a probability of $1/n$ and it could also be detected by user with a probability of $1/n$, which will ruin the reputation of P_j. So, it is reasonable to assume that P_j is honest in our scheme.

Security of Session Key: The session key $K_{ij} = \alpha^t \bmod N = (Z^e ID_j^{-1})^t \bmod N = w^k \bmod N$, if the adversary wants to acquire the session key, he need to compute $Z = g^k S_j \bmod N$. As r and k are random numbers chosen by the user U_i and the service provider P_j respectively, S_j is the private key owned by the P_j only, the adversary can't computes the session key K_{ij}.

Mutual Authentication: In our scheme, not only can the server verify the legal users, but the users can verify the legal service provider. We will show that our scheme achieves mutual authentication as follows.

No one can impersonate a legal user to request the service from the service provider in our new scheme SAIKA. Because of the hardness of discrete logarithms, it is computationally infeasible for the attacker to calculate the session key, without which it is hard for an attack to forge $(w, \{x_i\}, y, T)$. Therefore, the attacker can not establish a session key with service provider to authenticate himself as a legal user.

If an attacker wants to masquerade as the service provider, he should compute $Z = g^k S_j \bmod N$, so he need to get S_j, however, S_j is the private key of the service provider from SCPC, and it is nearly impossible for the attacker to know it. Therefore, we can say it is only the service provider that can authenticate itself to the user.

Forward Secrecy and **Backward Secrecy:** As t and k are both selected by the user and the service provider randomly, the session key of every period has no relation with each other. Therefore, if the session key generated in j period has been leaked, the attacker can't get any information of the session key generated before j period or after j period.

Our scheme can defeat replay attack. The messages transmitted over the network in our scheme can not be intercepted for reuse, because of the involvement of timestamp T and T'. And the server could check the freshness of a received message by testing whether the

transmission time is within legal transmission delay.

D. Comparison

Compared with Hsu-Chuang scheme, our scheme not only defends the private key disclosure attack of it, but also has some characters that it doesn't have.

In Hsu-Chuang scheme, if the user U_i wants to request a service from the service provider P_j, he only needs to send a service request, however, there exists a problem that the attacker may send lots of request to interfere other legal users' request. However, in SIAKA, if U_i wants to request a service, he should chooses a random number r and computes $C = r \oplus h(ID_j||T_0)$. As $h(.)$ is a one-way hash function and T_0 is a timestamp, the attacker can not send the service request randomly. Therefore, our scheme provides a way to defend illegal requests effectively.

Hsu-Chuang scheme requires that the identity of a user should be explicitly specified to facilitate authentication and further key exchange, which may violate users' privacy in some privacy sensitive applications, such as online drug store. However, in our new scheme, we address such issue. In SIAKA, the user U_i collects a group of n identities, and composes an identity list $L = \{ID_1, \ldots, ID_n\}$, including U_i's own identity $ID = ID_s$ ($s \in [1, n]$). U_i sends the n identities to the service provider P_j, so P_j does not know which the real identity of U_i is. Although P_j could guess it with the possibility of $1/n$, where n is large enough, the possibility will be very low. Introduction

In Hsu-Chuang scheme, the user's identity is specified to the service provider, but our scheme conceals the user's real identity in a group of identities. In order to confirm the user's identity, the user should reveal his identity to SCPC. In SIAKA, if P_j wants to confirm that U_i is the real user, he could send a request to SCPC, and SCPC will compare the received $\{x_i\}$ to verify U_i's real identity.

III. APPLICATION

As far as user privacy and anonymity is concerned, research on this topic usually focuses on two issues: anonymous communication and user anonymity [8]. Anonymous communication usually provides a communication channel that resists traffic analysis, so that the communicating parties can be anonymous against the eavesdroppers. A more complicated and seemingly paradoxical issue is user anonymity, which let the users hide their identities from the communicating peers. Here we use a concrete application of SAIKA to discuss user anonymity in the environment of distributed networks authentication.

A user Bob wants to download some files from a website Cindy; however, he doesn't want to disclose his identity. We assume that Alice is a trusted third party. The process can be described as follows.

(1) Alice uses a private key d to generate Bob's private key $\{S_i\}$ as $S_i = ID_i^d \bmod N$ ($ID_i \in L$, where L is a group of n identities collected by Bob) and Cindy's private key S_j as $S_j = ID_j^d \bmod N$. Then, Alice sends the private key S_i and S_j to Bob and Cindy respectively via a secure channel.

(2) Bob submits the service request to Cindy.

(3) After receiving Bob's service request, Cindy computes Z and sends Z to Bob.

(4) Bob computes the session key K_{ij} and encrypts L with K_{ij}. Finally, Bob sends the results to Cindy with the timestamp.

(5) Cindy computes K_{ij} and uses it to decrypt the received information, and then Cindy verifies the information received from Bob.

(6) Cindy sends the decrypted L to Bob with some change for verification.

(7) Bob verifies the results, if the equation holds, Bob trusts that Alice is the real vendor and establishes the connection with Cindy.

IV. CONCLUSIONS

In this paper, we exploited the weaknesses of a recently proposed user identification scheme with key preserving user anonymity for distributed computer networks (Hsu-Chuang scheme). That is, the service provider could get the user's private key, and then simulate the user to send the service request to other service providers. After that, we proposed a new scheme to overcome this limitation while achieving the same set of security properties. Then, we demonstrated the correctness of SIAKA and analyze the security properties of it. After that, we compared the differences of the two schemes. We also gave an example to illustrate the application of SIAKA.

Acknowledgment

This work is supported by the Jiangsu Provincial Natural Science Foundation of China (BK2007035), the open research fund of National Mobile Communications Research Laboratory, Southeast University (W200817) and the Science and Technology Foundation of CUMT (0D080309).

References

[1] W.B. Lee, C.C. Chang. User identification and key distribution maintaining anonymity for distributed computer network. Computer Systems Science and Engineering 15 (4) (1999) 113-116.

[2] T.S. Wu, C.L. Hsu. Efficient user identification scheme with key distribution preserving anonymity for distributed computer networks. Computers and Security 23 (2) (2004) 120-125.

[3] Y. Yang, S.Wang, F. Bao, J.Wang, R.H. Deng. New efficient user identification and key distribution scheme providing enhanced security. Computers and Security 23 (8) (2004) 697-704.

[4] K.Mangipudi, R. Katti. A secure identification and key agreement protocol with user anonymity (SIKA). Computers and Security 25 (6) (2006) 420-425.

[5] Chien-Lung Hsu, Yu-Hao Chuang. A novel user identification scheme with key distribution preserving user anonymity for distributed computer networks. Information Sciences 179 (2009) 422-429

[6] W.Diffie, M.Hellman. New directions in cryptography. IEEE Transactions on Information Theory IT-22 (6) (1976) 644-654.

[7] T.ElGamal. A public-key cryptosystem and a signature scheme based on discrete logarithms. IEEE Transactions on Information Theory IT-31 (4) (1985) 469-472.

[8] .Bo, Z., Wan, Z.G., Kankanhalli, M.S., Feng, B., Deng, R.H. Anonymous secure routing immobile ad-hoc networks, Local Computer Networks, 2004. 29th Annual IEEE International Conference on 16-18 Nov. (2004) 102-108.