A New Anonymous Ring Authenticated Key Exchange Protocol

Hui Cui, and Tianjie Cao
School of Computer, China University of Mining and Technology, Xuzhou, China
snager@163.com, tjcao@cumt.edu.cn

Abstract—Due to the growing importance in the fields of computer networks, how to keep user's privacy has drawn more and more concerns in recent years. In this paper, based on a ring signature scheme, we propose a new secure anonymous authenticated key exchange protocol, which inherits all the good virtues of the previous protocols. Furthermore, we discuss the security attributes of our new scheme. Finally, we give an extension of our new scheme's application and make a conclusion of this paper.

Index Terms—ring signature, anonymous authentication, key exchange

I. INTRODUCTION

Due to the growing importance in the application of computer networks [1], authentication supporting key exchange has been widely used to protect resources from illegal access. However, traditional authentication schemes supporting key exchange require that the identity of the user should be explicitly specified to facilitate authentication and further key exchange, which may violate the user’s privacy in some privacy-sensitive applications, such as online drug stores. In order to solve such seemingly paradoxical issue, many schemes have been proposed, and in this paper we also give a scheme to solve such problem.

A. Anonymity in Authentication Scheme

As far as user privacy and anonymity is concerned, research on this topic usually focuses on two issues: anonymous communication and user anonymity [2]. Anonymous communication [3] usually provides a communication channel to resist traffic analysis, so that the communicating parties can be anonymous against the eavesdroppers.

A more complicated and seemingly paradoxical issue is user anonymity, which aims at providing users anonymity when they are using the network by letting them hide their identity from the communicating peers. User anonymity existing in an anonymous authentication scheme [4] is a protocol that allows a member called a prover of a group to convince a verifier that he is a real member of the group without revealing any information about his identity. An immediate way to achieve user anonymity is to assign an alias name to each user, and every user will use his alias name to login to perform key exchange with the server, instead of using his real identity. However, such idea does not work since the server could always match user’s alias name with his real identity. In this paper we will discuss user anonymity in authentication scheme.

User anonymity is first addressed in the setting of digital signature with the introduction of group signature [5]. A group signature allows each group member to sign on behalf of the whole group anonymously without revealing his own identity. So the verifier of the group signature could not tell who is the real signer in the group. User anonymity is also achieved in ring signature proposed by Rivest, Shamir and Tauman [6], where a signer takes a list of public key of other people as input to sign a message, and a signature verifier could not spot the real signer in the public key list.

Both ring and group signatures share the same idea of achieving user anonymity, which is to hide one’s own identity among a group of identities. The difference between ring signature and group signature is that the group is formed more freely in the setting of ring signature [7]. In ring signature, the public key list which implicitly defines a group is composed by a signer himself without the help of a group manager, and the signer needs not to inform or interact with people involved in the list. While in the setting of group signature, a user must join the group first by interacting with the group manager to obtain a group membership. Therefore, it is more attractive to achieve user anonymity in a ring fashion, and we will use a ring signature scheme in our new protocol.

B. Related works

Authentication schemes that deal with anonymous issue are proposed by Chien.H.Y., Chen.C.H. [8] and Viet.D.Q., Yamamura.A. and Tanaka.H [9] respectively. However, the former indeed deals with anonymous communication not user anonymity, and the latter uses password tables at the server side and needs a lot of exponential operations. Jing Yang et al. [10] firstly point out the vulnerabilities of both Viet et al.’s and Shin et al.’s anonymous password-based authenticated key exchange protocols, and then propose a new anonymous password-based authenticated key exchange (JZH) protocol. Zhuchen Chai et al. [11] propose an efficient password-based authentication and key exchange (CHCL) scheme to preserve user privacy, and analyse the security requirements of their scheme. Both of them try to achieve user anonymity without using group or ring signature schemes, however, we find that their user anonymity both exist the security defect, namely, they can’t satisfy anonymity and the server can guess the user’s identity.
C. Requirements to Evaluate Anonymous Authentication and Key Exchange Protocol

We intend to design an efficient authentication scheme, which is expected to inherit all the good virtues of the previous schemes and support Diffie-Hellman key exchange protocol. Here we summarize all these requirements to evaluate our new scheme as follows.

User Anonymity: The scheme should preserve 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.

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

Forward Security: The scheme should satisfy forward security, namely, if the session key generated in j period has been leaked, the attacker can’t forge any session key generated before j period. Therefore, the scheme should defeat some attacks like replay attack and so on.

Reveal User’s Identity: The scheme should be able to reveal the user’s identity, namely, after the protocol, if the user wants to reveal his own identity, he can reveal it to the server. In some cases, the server may not believe that the user is the real user, so it is necessary for the user to reveal his identity to the server after the protocol.

II. PRELIMINARY

In this section, we review some fundamental backgrounds used throughout this paper.

A. Diffie-Hellman Key Exchange Protocol

In 1976, Diffie and Hellman proposed a well-known key exchange scheme, which allows two parties to negotiate a secret session key over insecure networks. The protocol works as follows:

1. Alice secretly chooses \( x \in \mathbb{Z}_p^* \) and sends \( X = g^x \mod p \) to Bob, where \( p \) is a large prime and \( g \) is a primitive element in \( \mathbb{Z}_p^* \).
2. Meanwhile, Bob secretly chooses \( y \in \mathbb{Z}_p^* \), and sends \( Y = g^y \mod p \) to Alice.
3. Alice computes \( K = X^y \mod p \). Bob computes \( K' = Y^x \mod p \).

After the protocol, Alice and Bob indeed share a secret session key \( K = K' = g^{xy} \mod p \).

B. Ring Authenticated Encryption Scheme

Rivest R. et al. [6] introduce the notion of ring signature, which makes it possible to specify a set of possible signers without revealing which member actually produced the signature. In their ring signature scheme, they define a family of keyed combining functions \( C_k (y_1, y_2, \ldots, y_n) \) which are still very useful in our new scheme. Every keyed combining function \( C_k (y_1, y_2, \ldots, y_n) \) takes as input the key \( k \), an initialization \( b \)-bit value \( v \), and arbitrary values \( y_1, y_2, \ldots, y_n \). Given any fixed values for \( k \) and \( v \), each such combining function uses \( E_k \) as a sub-procedure, and outputs a \( b \)-bit value \( z \), which has the following three properties:

1. For each \( s, 1 \leq s \leq n \), and for any fixed values of all the other inputs \( y_i, i \neq s \), the function \( C_k (y_1, y_2, \ldots, y_n) \) is a one-to-one mapping from \( y_s \) to the output \( z \).
2. For each \( s, 1 \leq s \leq n \), given a value \( z \) and the values for all inputs \( y_i \) except \( y_s \), it is possible to efficiently find a value \( y_s \) for such that \( C_k (y_1, y_2, \ldots, y_n) = z \).
3. Given \( k, v \), and \( z \), it is infeasible for an adversary to solve the equation \( C_k (y_1, y_2, \ldots, y_n) = z \) for \( x_1, x_2, \ldots, x_n \), if the adversary cannot invert any of the trap-door functions \( g_t (\cdot), g_t(\cdot)^*, \ldots, g_t(\cdot)^{(n)} \).

J. Lv et al. [12] propose a new authenticated encryption scheme called ring authenticated encryption scheme, which combines the two notations of ring signature and authenticated encryption together, and they also present a ring authenticated encryption scheme based on discrete logarithm problem. T. Cao et al. [13] found that L. V. et al.’s scheme doesn’t actually achieve signer-verifiability and recipient-verifiability properties, and they propose an improved scheme to eliminate the weaknesses.

III. THE NOVEL ANONYMOUS AUTHENTICATED KEY EXCHANGE PROTOCOL

In this section, based on T. Cao et al.’s improved ring signature authenticated encryption scheme [13], we propose a new anonymous authenticated key exchange protocol, NAAKE, as mentioned previously, which is secure against the above attack. Furthermore, we analyse the security of our new scheme and give an example of our scheme’s application.

A. Protocol Description

Before proceeding, we assume the existence of a publicly defined symmetric encryption algorithm \( E \) such that for any key \( k \), the function \( E_k \) is a permutation over \( b \)-bit strings. And we also assume the existence of a family of keyed combining functions \( C_k (y_1, y_2, \ldots, y_n) \) (defined in the ring signature scheme of Rivest et al. [4]) and a publicly defined collision-resistant hash function \( H(\cdot) \) that maps arbitrary inputs to strings of constant length, which are used as keys for \( C_k (y_1, y_2, \ldots, y_n) \).

Initialization: For the user \( U_i \) who wants to generate a session key with the server \( S \), he uses a ring of \( n \) logged-on users, and does follows.

Step 1. Chooses the following parameters: a large primes \( p_i \) such that it is hard to compute discrete logarithms in \( GF(p_i) \), another large prime \( q_i \) such that \( q_i | p_i - 1 \), a generator \( g_i \) in \( GF(p_i) \) with order \( q_i \).

Step 2. Chooses \( x_i \) \( \in Z_n^* \) as his private key, and computes the public key \( y_i = g_i^{x_i} \mod p_i \).

Step 3. Defines a trap-door function \( f(\alpha, \beta) = E_y (\cdot)^{\alpha} \cdot g_y ^{\beta} \mod p_i \); its inverse function \( f^{-1}(\cdot) \) is defined as \( f^{-1}(\cdot) = (\alpha, \beta) \), where \( \alpha \) and \( \beta \) are computed as follows (\( K_i \) is a random integer in \( Z_q^* \)).

\[
\alpha = y_i \cdot (E_{x_i} (\cdot)^{K_i} \mod p_i)^{\alpha} \mod p_i , \quad (1)
\]

\[
\alpha = a \mod q_i , \quad (2)
\]
\[ \beta = K \cdot (g^k \mod p) - x_q \cdot \alpha^* \mod q \]  
\[ \beta = K \cdot (g^k \mod p) - x_q \cdot \alpha^* \mod q, \]  

\( U_i \) makes \( p, q, g, \) and \( x_q \) public, and keeps \( x_q \) secret.

The server S chooses a large prime \( p \) such that it is hard to compute discrete logarithms in \( GF(p) \); another large prime \( q \) such that \( q \mid p - 1 \), a generator \( g \) in \( GF(p) \) with order \( q \), and a random integer \( x_B \) from \( Z_q \) as his private key, computes his public key \( y_B = g^{x_B} \mod p \), and publishes \( (y_B, p, q, g) \).

**Anonymous Authenticated Key Exchange:** This phase is initiated by \( U_i \), and ends in three rounds, resulting in an authenticated session key between \( U_i \) and \( S \).

**Round 1:** When the \( t \)th user \( U_t \) wants to generate a session key on the behalf of \( n \) ring members \( U_1, U_2, ..., U_n \), where \( 1 \leq t \leq n \), \( U_i \) does the following.

1. Chooses a random integer \( x_1, x_2 \) from \( Z_q^* \), computes
   \[
   R = g^{x_1} \mod p, \quad Q = y_B^{x_2} \mod q, \quad X = g^x \mod p, \quad V = X \cdot g^Q \mod p.  
   \]
   and computes \( l \) as \( l = H(X, Q, V, y_B, T) \).

2. Chooses a pair of values \((a_i, \beta_i)\) for every other ring member \( U_i \) \((1 \leq i \leq n, i \neq k)\) in a pseudorandom way, and computes \( y_i = f(a_i, \beta_i) \mod p \).

3. Picks randomly a \( b \)-bit initialization value \( v \), and solves out \( y_i \) from equation \( C_k, (\ y_1, y_2, ... , y_n = v \).

4. Computes \( \ (a_i, \beta_i) = f^{-1}(y_i) \) by using the trap-door information of \( f \); first, chooses a random integer \( K \in Z_q^* \), computes \( \alpha_i \) by (1), and keeps \( K \) secret; second, computes \( \alpha_i^* \) by (2); finally, computes \( \beta_i \) by (3).

5. The ring signature \( \sigma \) on \( X \) is \( U_i, U_2, ..., U_n, V, R, (a_i, \beta_i), (a_2, \beta_2), ..., (a_n, \beta_n) \).

   Finally, the user \( U_i \) sends \( \sigma \) and \( T \) to the server \( S \).

**Round 2:** \( S \) follows the following to recover and verify \( X \) from the signature \( \sigma \).

1. Computes \( Q = R^v \mod p \mod q, \) recovers \( X \) as \( X = V \cdot g^Q \mod p, \) and hashes \( \ X, Q, V, \) and \( y_B \) to recover \( l \) as \( l = H(X, Q, V, y_B, T) \).

2. Computes \( y_i = f(a_i, \beta_i) \mod p, \) for \( i = 1, 2, ..., n \).

3. Checks whether \( C_k, (y_1, y_2, ..., y_n) = v \). If it holds, \( S \) accepts \( X \) as valid (otherwise, \( S \) rejects \( X \)), and then does the following: chooses a random integer \( x_3 \) from \( Z_q^* \), and computes
   \[
   Y = g^{x_3} \mod p, \\
   K_s = X^{y_B} \mod p, \\
   h = H(K_s, X, Y, T), \\
   \]
   and then sends \( \{h, Y, T\} \) to \( U_i \).

**Round 3:** \( U_i \) verifies whether \( K_s \) is from the server \( S \).

\( U_i \) computes \( K_s' = Y^x \mod p, \) and hashes \( K, X, Y \) to get \( h' \) as \( h' = H(K_s', X, Y, T) \). If \( h' = h, \) \( U_i \) accepts \( K_s \) as the session key.

**User’s Identity Revealing:** If the actual user \( U_i \) is willing to disclose to the server \( S \) (or any third party Bob) that the signature is generated by him, then he does the following.

1. \( U_i \) computes \( M = g^k \mod p, \) and sends \( (M, y_A) \) to Bob \( ((M, y_A) \) and \( \sigma \) to the third party, respectively.)

2. Bob (the third party, respectively), who already knows \((a_i, \beta_i)\), computes \( \beta_3 = a_i \mod q, \) and checks \( M^{\beta_3} = f^{\beta_3}(y_B) \mod p, \) only if the equation holds will Bob (the third party, respectively) accepts that as the real signer of the signature.

**B. Security Discussion**

As we discussed earlier, our new anonymous authenticated key exchange scheme satisfies the following security requirements.

**User Anonymity:** In our new scheme, for a given signature on \( X \), any verifier can only be convinced that the ring signature is actually produced by at least one of the possible users. If the actual user does not reveal the seed \( K \), then any verifier cannot determine who is the actual user. The limited anonymity is computational and depends on the security of the pseudorandom generator. Anyone cannot determine the identity of the actual user in a ring of size \( n \) with probability greater than \( 1/n \). As \( k \) and \( v \) are fixed in a ring signature, there are \( 2^k \cdot (x_1, x_2, ..., x_n) \) satisfying the equation \( C_k, (y_1, y_2, ..., y_n) = v \), and the probability of the generation of \((x_1, x_2, ..., x_n)\) is the same, ignoring the subjectivity of the user, so the signature can’t leak the identity information of the user, namely it satisfies anonymity.

**Mutual Authentication:** In our scheme, not only can the server verify the legal users, but the users can verify the legal server. No one can impersonate a legal user inside a ring to login the server in our scheme. Because of the hardness of inverting hash function \( f() \), it is computationally infeasible for the attacker to calculate \( (a_i, \beta_i) \), without which it is hard for an attack to forge a signature. Therefore, the attacker cannot establish a session key with server to authenticate himself as a member within the submitted signature.

If an attacker wants to masquerade as the server, he should compute \( h = H(K_s, X, Y) \), so he need to get \( x_B \) in order to compute \( X \), however, \( x_B \) is the private key of the server, and it is nearly impossible for the attacker to know it. Therefore, we can say it is only the server that can authenticate itself to the user.

**Forward Security:** As \( x_A \) and \( x_B \) are both selected 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.

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. And the server could check the freshness of a received message by testing whether the transmission time is within legal transmission delay.

**Reveal User’s Identity:** In our scheme, after the protocol, if the user wants to confirm his identity, the actual user could disclose his identity to the server (or any
third party Bob). As $K$ is a secret number chosen by the user himself, only he can compute $M = g^k \mod p$. And Bob (the third party, respectively) who already knows $(a_0, b_0)$ can compute $a_0^* = a_0 \mod q_0$ and checks $M^{a_0^*} = g^{a_0^*} \cdot y_0^{-a_0} \mod p$. If it holds, Bob accepts that he is the real user of the message $X$. Thus, the user completes the revelation of his identity.

C. Application

It is apparent that a very important part of NAAKE is the generation of the ring. As it is the communication between the user and the server, a general method is collecting the registered users to generate the ring. However, if the number of the users is very large, the calculated amount will also be very large. In order to solve such a problem, we can use the role-based access control (RBAC) [16] technique. Suppose that all the users are divided into three types, general user, super user and administrator, and their rights to access the resources are different. And then, in order to reduce the calculation amount, we can do it as follows. Given that a super user wants to generate a session key $k_s$ with the server, he can generate a ring of the users belonging to the role of general user. Therefore, the ring number will reduced to the number of super users instead of all the users.

Through this method, when the number of the user is very large, the calculation amount of the user would be greatly reduced. As the server doesn’t know the user’s role, the probability for him to guess the user’s identity is still $1/n$, where $n$ is the number of all the user.

IV. CONCLUSION

User anonymity is attached great importance in recent years to preserve user privacy in wired or wireless network environments. In this paper, we proposed a new anonymous authenticated key exchange protocol, NAAKE. Furthermore, we analyse the security of our new scheme and make an extension of this new scheme’s application.

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