An Authentication Protocol Without Trusted Third Party

Shiuh-Pyng Shieh, *Member, IEEE*, Wen-Her Yang, and Hun-Min Sun

***Abstract*— A secure authentication protocol which supports both the privacy of messages and the authenticity of communicating parties is proposed. The trusted third party (key information center) is not needed once the secure network system is set up. Mutual authentication and key distribution can be achieved with two messages merely between two parties involved.**

***Index Terms*—Authentication protocol, ID-based scheme.**

I. Introduction

THE first ID-based scheme, proposed by Shamir [3], supports only digital signature rather than message encryption. Tsujii proposed another ID-based cryptosystem based on the discrete logarithm problem [4], which suffers from the conspiracy problem, and needs high overhead of exponential computations. Okamoto and Tanaka extended Shamir's idea and combined digital signature and key distribution in a simple ID-based scheme [2] which supports message encryption and withstands the conspiracy problem. However, in the scheme user identifications may be forged, user secret information may be disclosed, and the high overhead of exponential computations is needed.

In this letter, we propose a new authentication protocol in which the key information center is needed only when the secure network system is being set up or when new users request to register. Not only does our protocol need fewer exponential computations but it also resolves the security problems that appeared in the Okamoto and Tanaka's scheme.

II. Secure Authentication Protocol

Both the ID-based scheme and symmetric cryptographic technique are used in the new secure authentication protocol. The ID-based scheme is used for system setup and authentication, while the symmetric cryptographic is used for subsequent message encryption to obtain better communication performance. There are two phases in the new authentication protocol. The *initial phase* is completed at the key information center to set up the system, and the *authentication phase* is executed between the two communication parties to achieve mutual authentication and exchange the common session key.

Manuscript received November 20, 1996. The associate editor coordinating the review of this letter and approving it for publication was Dr. C. Dooligeris.

The authors are with the Department of Computer Science and Information Engineering, College of Electrical Engineering and Computer Science, National Chiao-Tung University, Hsinchu, Taiwan 30010; (email: ssp@csie.nctu.edu.tw).

Publisher Item Identifier S 1089-7798(97)04328-7.

A. Initial Phase

The information center is responsible neither for mutual authentication nor for the generation of common keys. The role of this center is to simply generate public and secret information for newly registered users. When the secure network system is setting up, the key information center will execute the following steps.

1) Choose two large prime numbers p and q, and let $n = p \cdot q$.
2) Obtain the center's secret information d from the following computation, where d is only known by the center.

$$3 \cdot d \,(\mathrm{mod}(p-1)\cdot(q-1)) = 1. \tag{1}$$

3) Find an integer g which is a primitive element in both $GF(p)$ and $GF(q)$, where g is the center's public information.
4) Let ID_i denote the identity of user i who requests to register to this secure network. ID_i could be composed of name, address, ..., and so on.
5) Choose a one-way function f to compute the extended identity (EID_i) of i as follows

$$\begin{aligned}\mathrm{EID}_i &\equiv f(ID_i) \pmod{2^N} \\ &\equiv (\mathrm{EID}_{i1}, \mathrm{EID}_{i2}, \ldots, \mathrm{EID}_{iN})\end{aligned} \tag{2}$$

where N denotes the bit length of EID.
6) After computing EID_i, calculate the user secret information S_i as

$$S_i \equiv \mathrm{EID}_i^d \pmod{n}. \tag{3}$$

From the relations above, the following equation would be obtained.

$$\mathrm{EID}_i \equiv S_i^3 \pmod{n}. \tag{4}$$

7) Send $(n, g, f(x), S_i)$ back to user i over a secure channel, such as a certified and sealed mail. Upon receipt of the information, user i must keep S_i secret and store the public information $(n, g, f(x))$.

Once the secure network system is set up, the key information center is not needed except when new users join. The center's secret information d must be stored secretly for subsequent use. However, the integers p and q will be no longer used and should be thrown away secretly. When a new user requests to join, he sends the center his ID. Upon receipt of the user ID, the center repeats steps 5–7.

B. Authentication Phase

The new authentication protocol only needs two messages to complete the mutual authentication. Upon receipt of the first message from user i, user j verifies the message contents. If the verification succeeds, he believes that the message is sent by user i. Thus user j authenticates user i. Similarly, user i authenticates user j with the second message. The execution steps for mutual authentication and key exchange for a session are listed as follows.

1) If user i wishes to communicate with user j, he generates a random number ri and calculates the following two integers:
$$X_i \equiv g^{3 \cdot ri} \pmod{n} \quad (5)$$
$$Y_i \equiv S_i \cdot \text{time}_i \cdot g^{2 \cdot ri} \pmod{n} \quad (6)$$
where *time*$_i$ is the time he calculated the two integers.
2) User i sends these two integers X_i and Y_i together with *ID*$_i$ and time_i to user j.
3) Upon receipt of the message, User j compares time_i with the present local time. If the difference between time_i and the present local time is shorter than the valid period, the message received is considered valid. According to different network environments, the length of the valid period can be adjusted. (In order to avoid valid messages being rejected in a network where clocks are not at least loosely synchronized, the step for comparing time_i and present local time should be skipped.) Then, user j calculates $\text{EID}_i = f(ID_i)$ and checks whether the following equation holds:
$$\text{EID}_i \cdot \text{time}_i^3 = Y_i^3 / X_i^2 \quad (7)$$
4) If the equation holds, user j believes the message is sent by user i and keeps X_i for generation of the common key later. Then, he generates a random number rj and calculates the following two integers:
$$X_j \equiv g^{3 \cdot rj} \pmod{n} \quad (8)$$
$$Y_j \equiv S_j \cdot \text{time}_j \cdot g^{2 \cdot rj} \pmod{n}. \quad (9)$$
5) User j sends these two integers X_j and Y_j along with *ID*$_j$, and time_i to user i.
6) Upon receipt of the message, user i checks whether time_i is identical to the one he sent. (time_i herein can be considered as a nonce of user i, which is only used for once.) If yes, he calculates $\text{EID}_j = f(ID_j)$ and checks if the following equation holds:
$$\text{EID}_j \cdot \text{time}_j^3 = Y_j^3 / X_j^2. \quad (10)$$
7) If it is true, user i calculates the session key K_{ij} as follows:
$$K_{ij} = X_j^{ri} = g^{3 \cdot ri \cdot rj} \quad (11)$$
8) In the same way, user j calculates the session key K_{ji} as follows:
$$K_{ji} = X_i^{rj} = g^{3 \cdot ri \cdot rj} \quad (12)$$
9) Users i and j use $K_{ij} = K_{ji}$ as the common key of this session to encrypt the communicating messages.

III. Computation Overhead

In the Okamoto and Tanaka's scheme, each party needs five exponential computations to complete mutual authentication and exchange a common key for each session (one for X_i, one for Y_i, two for equation check, and one for the common key calculation).

Our protocol reduces the number of exponential computations for each communication session from five to two. In the *authentication phase*, we can first compute g^{ri}, then calculate X_i and Y_i as follows:
$$X_i \equiv g^{ri} \cdot g^{ri} \cdot g^{ri} \pmod{n} \quad (13)$$
$$Y_i \equiv S_i \cdot \text{time}_i \cdot g^{ri} \cdot g^{ri} \pmod{n} \quad (14)$$
No exponential computation but multiplication is needed in these two equations. The verification of sender's identity [see (7)] can also be accomplished without exponential computation in the same way. Therefore, our protocol needs only two exponential computations (one for g^{ri}, and one for common key $(X_i)^{ri}$).

IV. Security Analysis

Our protocol provides message encryption and the authenticity of communicating parties to guarantee the privacy and security of network communication. It does not have the conspiracy problem existing in the Tsujii's scheme because its security relies on the difficulty of computing the discrete logarithm problem. If a forger wants to masquerade user i to communicate with others, he must find two integers x and y satisfying the following equation:
$$y^3 = \text{EID}_i \cdot \text{time}_i^3 \cdot x^2. \quad (15)$$
The use of low public exponents in this equation does not lower the the difficulty to crack (y, x). Although the forger can get a pair of integers (y^3, x^2) that makes the equation hold, the pair (y, x) is unattainable because computing (y, x) pair from (y^3, x^2) is a discrete logarithm problem.

Our protocol can also protect users from the Hastad's attack. Hastad proposed an attack on using RSA with low exponents in a public key network [1]. To illustrate this attack, suppose that a message m is broadcasted to three parties in which the public exponents are $e_1 = e_2 = e_3 = 3$, and in which the moduli are n_1, n_2, and n_3. The encrypted messages are
$$m^3 \bmod n_1, \quad m^3 \bmod n_2, \quad m^3 \bmod n_3.$$
Using the Chinese remainder theorem, one can find $m^3 \bmod n_1 n_2 n_3$. However, $m^3 < n_1 n_2 n_3$ because $m < n_1, n_2, n_3$. Therefore, m^3 is not affected by being reduced modulo $n_1 n_2 n_3$, and the message can be recovered by taking the cube root of m^3. This attack will not succeed in our protocol, because the same modulus n is used for all parties.

Although we use a timestamp to check the message legality, the replay-attack will not succeed in our protocol, even if the assumption of synchronized clocks does not exist. Considering the following scenario, an intruder eavesdropped a communication session, e.g., the communication between user i and

user j. The intruder may replay an old authentication message captured in the old session. Upon receipt of the old messages, user j checks the legality of time_i. If the system's clock time is synchronized, he knows the message is invalid by examining time_i and therefore discards this authentication message. If the system's clock is not synchronized, he may reconsider the message. Then he chooses a new random number rj' and replies the following message to the intruder:

$$X_j' \equiv g^{3 \cdot rj'} \pmod{n} \tag{16}$$

$$Y_j' \equiv S_j \cdot \text{time}_i \cdot g^{2 \cdot rj'} \pmod{n}. \tag{17}$$

However, the common key of this communication session is $K' = g^{3 \cdot ri \cdot rj'}$, instead of the old common key $K = g^{3 \cdot ri \cdot rj}$. The intruder cannot compute the new common key K' without knowing the random number ri. Since the old messages are all encrypted by the old common key K, he cannot successfully replay the old messages he eavesdropped. User j may try to decrypt them by the new common key K', but the decryption fails. Consequently, he closes the connection and the attack fails.

Our protocol also does not have the two weaknesses appeared in the Okamoto and Tanaka's scheme.

(1) Our protocol uses two small prime numbers 3 and 2 instead of the two integers e and c in the Okamoto's scheme. Since the possibility no longer exists that e may be a factor of c, user secret information will not be disclosed in our protocol.

(2) The attack of forged authentication messages will fail in our protocol because of the one-way function $f(x)$. If a malicious user wants tosend a forged message to user j, he will randomly choose a pair numbers (X', Y'). Although a bogus EID$'$ may be computed from (15), he cannot derive the correct *ID*$'$ from EID$'$ because of the one-way function $f(x)$. If he randomly chooses an identity information *ID*$''$, and sends it together with (X', Y'), the time he wrote the message, and the forged message, upon receiving the packet, user j will get EID$''$ from $f(ID'')$, instead of EID$'$. Consequently, the verification of (7) will fail, and user j will reject the forged request. Therefore, our protocol is able to protect user communication from the attack of forged requests.

V. Conclusion

An ID-based authentication protocol is proposed in which both the key in formation center and files for the storage of public information are not required. Once the secure network system is set up, the authentication and key exchange can be handled solely by the two parties involved, instead of the key information center. This protocol resolves the problems appeared in the Okamoto and Tanaka's scheme. Even if the system clocks are not synchronized, it can withstand the replay problem. In contrast to five exponential computations needed in the Okamoto and Tanaka's scheme, our protocol needs only two exponential computations for mutual authentication and key exchange, thereby greatly reducing the load on communication devices.

References

[1] J. Hastad, "On using RSA with low exponent in a public key network," in *Lecture Notes in Computer Science: Advances in Cryptology-CRYPTO'85 Proc.*, pp. 403–408.

[2] E. Okamoto and K. Tanaka, "Identity-based information security management system for personal computer networks," *IEEE J. Select. Areas Commun.*, vol. 7, pp. 290–294, Feb. 1989.

[3] A. Shamir, "Identity-based cryptosystems and signature schemes," in *Proc. Crypto-84*, Santa Barbara, CA, 1984, pp. 47–53.

[4] S. Tsujii, T. Itho, and K. Kurosawa, "ID-based cryptosystem using discrete logarithm problem," *Electron. Lett.*, vol. 23, pp. 1318–1320, Nov. 1987.