Security of Improved Identity-based Conference Key Distribution Systems

Kenji Koyama Kazuo Ohta*

Basic Research Laboratories
Nippon Telegraph and Telephone Corporation
3-9-11, Midori-cho, Musashino-shi, Tokyo, 180 Japan

*Communications and Information Processing Laboratories
Nippon Telegraph and Telephone Corporation
1-2356, Take, Yokosuka-shi, Kanagawa, 238-03 Japan

Abstract

At Crypto-87 conference, we proposed identity-based key distribution systems for generating a common secret conference key for two or more users. Protocols were shown for three configurations: a ring, a complete graph, and a star. Yacobi has made an impersonation attack on the protocols for the complete graph and star networks. This paper proposes improved identity-based key distribution protocols to counter his attack.

1. Introduction

Identity-based cryptosystems can simplify key management in cryptosystems. Shamir and Fiat proposed identity-based signature schemes [1, 2], and Okamoto proposed an identity-based scheme [3] for a public key distribution system [4]. In these schemes for two users, messages among users are authenticated using each user's identification information. If two or more users want to hold a conference, they must derive one common secret communication key for each link in the network. This common key for m (≥ 2) users is called a conference key. Ingemarsson et al. [5] presented a conference key distribution system (CKDS) with no authentication, where users are connected in a ring network. At the Crypto-87 conference, we [6] proposed an identity-based system for generating a conference key with authentication, called an identity-based conference

C.G. Guenther (Ed.): Advances in Cryptology - EUROCRYPT '88, LNCS 330, pp. 11-19, 1988.

key distribution system (ICKDS). Protocols in ICKDS were shown for three configurations: ring (Type-1), complete graph (Type-2), and star (Type-3). Yacobi [7] has made an impersonation attack on the Type-3. His attacking method can be generalized to Type-2. This paper proposes improved identity-based key distribution protocols to counter his attack. The previous protocol can detect a uni-directional attack and it cannot detect a bi-directional attack. However, the new protocol can detect both the uni-directional attack and the bi-directional attack. In Section 2, revised protocols of Type-2 and Type-3 are described, clarifying the difference between the previous and new versions. In Section 3, Security for these protocols is discussed. Details of the attack by Yacobi are stated, and it is shown that our improvement resolves the problem.

2. Improved ICKDSs

All ICKDSs are implemented in two phases: the first phase is carried out at a trusted center, and the second phase at each user's location. During the first phase, the trusted center generates a secret system key, a public system key, and secret user keys with users' identification information. The secret system key is known only to the center. The public system key is common to all users. Each secret user key, which is transmitted through secure channel such as smart card, is known only to each user and the center. Once the first phase is carried out, the second phase can be repeated to generate a different conference key. In the second phase, no further interaction with the center is required either to generate a key or to verify proofs of identity.

For simplicity, only improved protocols in a complete graph (Type-2) and in a star (Type-3) are shown in Subsections 2.1 and 2.2, respectively.

During the first phase of Type-2 and Type-3, the center generates three large primes p, q, and r, and the partial product $n = pq$. It determines integers $(e,\ d)$ in a way similar to that of the RSA cryptosystem [8]:

$$ed \equiv 1 \pmod{L}, \qquad L = \mathrm{lcm}\,((p-1),\ (q-1),\ (r-1)), \tag{2.1}$$

where e is a prime such that $nr/2 < e < nr$. Note that every integer in $[1,\ nr]$ except e is coprime to e. The center also determines an integer g which is a primitive element over $GF(p)$, $GF(q)$, and $GF(r)$. Note that g is easily generated while the factors of $(p-1)$, $(q-1)$, and $(r-1)$ are known. For user i whose identification information is I_i, the center calculates integer S_i:

$$S_i = I_i^d \bmod nr. \tag{2.2}$$

Note that $I_i = S_i^e \bmod nr$. As a result, the center generates a secret system-key (p, q, d), a public system-key (n, r, g, e), and a secret user-key S_i for user i.

2.1 Improved protocol in a complete graph (Type-2)

During the second phase of Type-2, the conference key is generated and simultaneously distributed among m users. Users are connected in a complete graph network so that they always send messages to all other users. The key generation algorithm is the same for each user. For convenience, the procedure for two typical users, labeled i and j ($1 \leq i, j \leq m$, $i \neq j$), can be described as follows:

[Protocol]

step 1: User i chooses a random number P_i that is coprime to $(r-1)$. He computes $\overline{P}_i$:

$$P_i\overline{P}_i \equiv 1 \pmod{(r-1)}, \tag{2.3}$$

and keeps P_i and $\overline{P}_i$ secret. He then sends (X_i, Y_i):

$$X_i = g^{eP_i} \bmod nr, \tag{2.4}$$

$$Y_i = S_i g^{X_i P_i} \bmod nr, \tag{2.5}$$

to user j.

step 2: User j receives (X_i, Y_i). He checks whether the following $(m-1)$ congruences hold:

$$\frac{Y_i^e}{X_i^{X_i}} \equiv I_i \pmod{nr}, \tag{2.6}$$

If (2.6) holds, user j can verify that the message came from user i. User j chooses a secret random number R_j. He then sends (A_{ji}, B_{ji}):

$$A_{ji} = X_i^{eR_j} \bmod nr, \tag{2.7}$$

$$B_{ji} = S_j X_i^{A_{ji}R_j} \bmod nr, \tag{2.8}$$

to user i.

step 3: User i receives $(A_{ji},\ B_{ji})$. He checks whether the following $(m-1)$ congruences hold:

$$\frac{B_{ji}^{e}}{A_{ji}^{A_{ji}}} \equiv I_j \pmod{nr}, \tag{2.9}$$

If (2.9) holds, user i can verify that the message came from user j. He then computes conference key K_i:

$$K_i = (\prod_{j=1}^{m} A_{ji})^{\overline{P}_i} \bmod r. \tag{2.10}$$

The value of K_i $(1 \le i \le m)$ is the same for all users, because

$$K_i = g^{e^2(P_iR_1+P_iR_2+\ldots+P_iR_m)\overline{P}_i} \bmod r = g^{e^2(R_1+R_2+\ldots+R_m)} \bmod r. \tag{2.11}$$

Remarks:

(1) The exponent terms X_i in (2.5) and (2.6) and A_{ji} in (2.8) and (2.9) in this version were expressed by a constant c in the previous version [6]. This improvement makes Yacobi's attack on Type-2 and Type-3 ineffective. Details will be discussed in Section 3.

(2) Since e is chosen such that $nr/2 < e < nr$, X_i and A_{ji} are coprime to e with the probability $1-1/nr$ (≈ 1). This property in the improved version inherits from the previous version, where c is coprime to e. This property has effect of countermeasure on some attacks other than Yacobi's attack.

(3) The previous protocol [6] contained check congruences such as $Z_{ij} \equiv X_i^{U_j} \pmod{n}$, $C_{ij} \equiv A_{ji}^{V_j} \pmod{n}$, and related computations. The purpose of such congruences was to detect a uni-directional impersonation attack [6] other than Yacobi's attack. These check congruences and related computations are omitted in the new protocol because the new protocol can detect such attack in addition to Yacobi's attack.

2.2 Improved protocol in a star (Type-3)

Type-2 can be simplified by restricting the process so that $j = 1$ and $2 \le i \le m$. Therefore, users are connected in a star network so that messages are transmitted between user 1 and user i $(2 \le i \le m)$. In this simplified scheme called Type-3, we assume that user 1 collects and delivers messages. Without

loss of generality, this "center user" can be arbitrarily selected from among m users.

The improved protocol during the second phase of Type-3 is similar to that of Type-2. Note that user 1 can compute conference key $K_1 = g^{e^2 R_1}$ at any time. User i $(2 \leq i \leq m)$ computes conference key K_i at step 3 by:

$$K_i = A_{1i}^{\overline{P}_i} \bmod r. \tag{2.12}$$

The values of K_i $(2 \leq i \leq m)$ and K_1 are the same for all users, because

$$K_i = g^{e^2 R_1 P_i \overline{P}_i} \bmod r = g^{e^2 R_1} \bmod r. \tag{2.13}$$

Note that the value of conference key in Type-3 is dependent on only user 1's secret key R_1, while the value of conference key in Type-2 is equally dependent on each user's secret random number R_i.

3. Security

The security of the proposed systems is based on the difficulty of deriving secret information such as $(p,\ q,\ d,\ S_i,\ P_i,\ \overline{P}_i,\ R_i,\ K_i)$ in Type-2 and Type-3 from public keys, transmitted messages, and other user's secret keys. Secrecy of $(p,\ q,\ d,\ S_i)$ is based on the difficulty of factoring a large number n. Secrecy of $(P_i,\ \overline{P}_i,\ R_i,\ K_i)$ is based on the difficulty of computing discrete logarithm over $GF(r)$. Considering the best known algorithms for factoring $n = pq$ [9] and computing the discrete logarithm over $GF(r)$ [10], a designer can choose the size of p, q, and r. From the security viewpoint, the size of p and q should be at least 256 bits long, and the size of r should be at least 512 bits long.

The secrecy of the above secret keys is believed to be ensured in the previous version and the new version. However, the authenticity of the previous version has been partly broken by Yacobi's impersonation attack because it had weak points. The new version described in this paper realizes protocols to detect his attack. In this section, a summary of his attacking method and the effect of our countermeasures are shown.

3.1 Yacobi's bi-directional attack [7]

By extending our uni-directional attack [6], Yacobi [7] showed a bi-directional real time attack between user i and user j in Type-3 ($2 \leq i \leq m,\ j = 1$). Note that his attacking method can be generalized to Type-2 ($1 \leq i, j \leq m$). Since the attacker can hold both a correct key and a false key, this bi-directional impersonation attack would be successful in the previous protocol.

We summarize the generalized Yacobi's attack on the previous version where the constant term c was used instead of variable exponents X_i and A_{ji}. An attacker cuts the link between user j (or "center user" in the star) and user i. He mediates every communication between them. When communicating with user j he pretends to be user i (denoted by $\widetilde{i}$), and when communicating with user i he pretends to be user j (denoted by $\widetilde{j}$). First, the attacker chooses random P', and computes its inverse $\overline{P}'$ modulo $r-1$. He also computes the inverse of e (denoted by $\overline{e}$) modulo $r-1$. For step 1, the attacker eavesdrops the message $(X_i,\ Y_i)$ from user i to user j. Using the Chinese remainder theorem, he computes $(\widetilde{X}_i,\ \widetilde{Y}_i)$ modulo nr satisfying:

$$\begin{cases} \widetilde{X}_i \equiv X_i \pmod{n}, \\ \widetilde{X}_i \equiv g^{eP'} \pmod{r}, \end{cases} \qquad \begin{cases} \widetilde{Y}_i \equiv Y_i \pmod{n}, \\ \widetilde{Y}_i \equiv (I_i g^{ceP'})^{\overline{e}} \pmod{r}, \end{cases} \tag{3.1}$$

and sends the modified message $(\widetilde{X}_i,\ \widetilde{Y}_i)$ to user j. For step 2, user j verifies

$$\frac{\widetilde{Y}_i^e}{\widetilde{X}_i^c} \equiv I_i \pmod{nr}. \tag{3.2}$$

User j computes $(\widetilde{A}_{ji},\ \widetilde{B}_{ji})$:

$$\widetilde{A}_{ji} = \widetilde{X}_i^{eR_j} \bmod nr,$$

$$\widetilde{B}_{ji} = S_j \widetilde{X}_i^{cR_j} \bmod nr.$$

and sends it to user i. The attacker intercepts this communication. He chooses some random number $\widetilde{R}_j$. Using the Chinese remainder theorem, he computes $(\widehat{A}_{ji},\ \widehat{B}_{ji})$ modulo nr satisfying:

$$\begin{cases} \widehat{A}_{ji} \equiv \widetilde{A}_{ji} \pmod{n}, \\ \widehat{A}_{ji} \equiv X_i^{e\widetilde{R}_j} \pmod{r}, \end{cases} \qquad \begin{cases} \widehat{B}_{ji} \equiv \widetilde{B}_{ji} \pmod{n}, \\ \widehat{B}_{ji} \equiv (I_j (X_i)^{ce\widetilde{R}_j})^{\overline{e}} \pmod{r}, \end{cases} \tag{3.3}$$

and sends the modified message $(\widehat{A}_{ji},\ \widehat{B}_{ji})$ to user i. For step 3, user i verifies

$$\frac{\widehat{B}_{ji}^{e}}{\widehat{A}_{ji}^{c}} \equiv I_j \pmod{nr}. \tag{3.4}$$

Finally, user i creates session key:

$$\widetilde{K}_i = (\prod_{j=1}^{m} \widehat{A}_{ji})^{\overline{P}_i} \bmod r = g^{e^2(\widetilde{R}_1+\widetilde{R}_2+\ldots+\widetilde{R}_m)} \bmod r. \tag{3.5}$$

Using $\widetilde{R}_j$ $(1 \leq j \leq m)$, attacker $\widetilde{j}$ creates the session key:

$$\widetilde{K}_j = g^{e^2(\widetilde{R}_1+\widetilde{R}_2+\ldots+\widetilde{R}_m)} \bmod r. \tag{3.6}$$

Note that $\widetilde{K}_i = \widetilde{K}_j$. Therefore, this attack succeeds if the attacker mediates every communication between user i and user j.

For Type-3, where user $j(=1)$ is a center user, user i finally creates session key:

$$\widetilde{K}_i = \widehat{A}_{1i}^{\overline{P}_i} \bmod r = g^{e^2\widetilde{R}_1} \bmod r. \tag{3.7}$$

Using $\widetilde{R}_1$, attacker $\widetilde{1}$ creates the session key:

$$\widetilde{K}_1 = g^{e^2\widetilde{R}_1} \bmod r. \tag{3.8}$$

User 1, who is center user, creates session key:

$$K_1 = g^{e^2 R_1} \bmod r. \tag{3.9}$$

Using $\overline{P}_i'$, attacker $\widetilde{i}$ creates the session key:

$$K_i = \widetilde{A}_{1i}^{\overline{P}_i'} \bmod r = g^{e^2 R_1} \bmod r. \tag{3.10}$$

Note that $\widetilde{K}_i = \widetilde{K}_1$ and $K_1 = K_i$ $(2 \leq i \leq m)$. This attack on Type-3 is more realistic than that on Type-2 because it requires that the attacker manipulates only one link from user i $(2 \leq i \leq m)$ to user 1.

3.2 Improved protocol's effect against the Yacobi's attack

Note that the exponent terms X_i and A_{ji} in this improved protocol were expressed by a constant c in the previous protocol [6]. This improvement

makes Yacobi's attack on Type-2 and Type-3 ineffective. In the improved protocol, if an attacker adopts Yacobi's attack, ID checks mod nr in (2.6) and (2.9) (or (3.2) and (3.4)) do not pass. Since the purpose and function of (2.6) and (2.9) is the same, the case for (2.6) is described as an example. Consider the congruence (2.6) modulo nr by separating it into a congruence modulo n and a congruence modulo r. A check congruence modulo n in (2.6) is not satisfied because

$$\frac{\widetilde{Y}_i^e}{\widetilde{X}_i^{\widetilde{X}_i}} \equiv \frac{(S_i g^{X_i P_i})^e}{(g^{eP_i})^{\widetilde{X}_i}} \equiv I_i g^{P_i e(X_i - \widetilde{X}_i)} \not\equiv I_i \pmod{n}. \tag{3.11}$$

Therefore, (3.11) results in

$$\frac{\widetilde{Y}_i^e}{\widetilde{X}_i^{\widetilde{X}_i}} \not\equiv I_i \pmod{nr}. \tag{3.12}$$

Note that a check congruence modulo r in (2.6) is satisfied because

$$\frac{\widetilde{Y}_i^e}{\widetilde{X}_i^{\widetilde{X}_i}} \equiv \frac{(I_i g^{\widetilde{X}_i e P'})^{\overline{e}e}}{g^{\widetilde{X}_i e P'}} \equiv I_i \pmod{r}. \tag{3.13}$$

Similarly to (3.12), we have

$$\frac{\widehat{B}_{ji}^e}{\widehat{A}_{ji}^{\widehat{A}_{ji}}} \not\equiv I_j \pmod{nr}, \tag{3.14}$$

Therefore, the Yacobi's bi-directional attack becomes detectable.

4. Conclusion

Security has been improved in the new protocol with the variable exponents. That is, the improved protocol counters Yacobi's attack. The change of exponent terms has the same effect as the additional check congruences in the previous version. By deleting such additional check congruences, transmission efficiency is also improved in the new protocols. This is a side effect of improving security.

Acknowledgement

We would like to thank Dr. Yacov Yacobi for his nice attack on our previous version.

References

[1] SHAMIR, A. :"Identity-based cryptosystems and signature schemes", Proceedings of Crypto'84, Lecture Notes in Computer Science no. 196, Springer-Verlag, 1985, pp.47-53.

[2] FIAT, A. and SHAMIR, A. :"How to prove yourself: Practical solutions to identification and signature problems", Proceedings of Crypto'86, Lecture Notes in Computer Science no. 263, Springer-Verlag, 1987, pp.186-194.

[3] OKAMOTO, E.:"Proposal for identity-based key distribution systems", *Electron. Lett.*, 1986, **22**, pp.1283-1284.

[4] DIFFIE, W., and HELLMAN, M. E. :"New directions in cryptography", *IEEE Trans.* 1976, **IT-22**, pp.644-654.

[5] INGEMARSSON, I, TANG, D. T. and WONG, C. K. :"A conference key distribution system", *IEEE Trans.* 1982, **IT-28**, pp.714-720.

[6] KOYAMA, K. and OHTA, K. :"Identity-based conference key distribution systems", Proceedings of Crypto'87, Lecture Notes in Computer Science no. 293, Springer-Verlag, 1988, pp.175-184.

[7] YACOBI, Y. :"Attack on the Koyama-Ohta identity-based key distribution scheme", Proceedings of Crypto'87, (presented at the rump session), Lecture Notes in Computer Science no. 293, Springer-Verlag, 1988, pp.429-433.

[8] RIVEST, R. L., SHAMIR, A., and ADLEMAN, L.:"A method for obtaining digital signatures and public-key cryptosystems", *Commun. ACM,* 1978, **21**, pp.120-126.

[9] LENSTRA, Jr. H. W. :"Factoring integers with elliptic curves", preprint, May 1986

[10] COPPERSMITH, D., ODLYZKO, A. M. and SCHROEPPEL, R. :"Discrete logarithms in GF(p)" *Algorithmica* 1986, **1**, pp.1-15.