Identity-based conference key distribution systems

Kenji Koyama Kazuo Ohta*

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

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

Abstract : This paper proposes identity-based key distribution systems for generating a common secret conference key for two or more users. Users are connected in a ring, a complete graph, or a star network. Messages among users are authenticated using each user's identification information. The security of the proposed systems is based on the difficulty of both factoring large numbers and computing discrete logarithms over large finite fields.

1. Introduction

Shamir first proposed the idea of identity-based cryptosystems [1], in which each user's public key is his identification information such as his name, address, etc. The systems do not require any key directories. Therefore, 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, messages among users are authenticated using each user's identification information.

A two-user secret common key with authentication can be generated by the Shamir scheme [1], Fiat-Shamir scheme [2], Okamoto scheme [3] and so on. 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 among $m(\geq 2)$ users is called a conference key. Ingemarsson et al. [5] presented a conference key distribution system (CKDS) without authentication, where users are connected in a ring network. The purpose of this paper is to propose an identity-based system for generating a conference key with authentication. We call this system an identity-based conference key distribution system or ICKDS. We propose three types of identity-based conference key distribution systems, which are Type-1 in a ring network [6], Type-2 in a complete graph network [7], and Type-3 in a star network [7].

C. Pomerance (Ed.): Advances in Cryptology - CRYPTO '87, LNCS 293, pp. 175-184, 1988.

All ICKDSs are carried out in two phases: the first phase is carried out at a trusted center, and the second phase at each user's location. 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. Protocols of Type-1, Type-2, and Type-3 for generating a conference key among m users are described in Sections 2, 3, and 4. Security and transmission efficiency for these protocols are discussed in Sections 5 and 6.

2. ID-based CKDS in a ring (Type-1)

2.1 First phase

During the first phase of Type-1, the center generates two large primes p, q and the product $n = pq$. It determines integers $(e,\ d)$ in a way same as that of the RSA cryptosystem [8]:

$$ed \equiv 1 \pmod{L}, \qquad L = \text{lcm}\ ((p-1),\ (q-1)), \qquad 3 \leq e,\ d < L, \tag{2.1}$$

where lcm denotes a least common multiplier and e is coprime to L. It also determines a prime c $(3 \leq c < L)$, and an integer g which is a primitive element over both $GF(p)$ and $GF(q)$. Let M be the largest number of expected conference members. For user i whose identification information is I_i, the center calculates integers S_i:

$$S_i = I_i^h \bmod n, \qquad h = d^{M-1} \bmod L. \tag{2.2}$$

Note that $I_i = S_i{}^{e^{M-1}} \bmod n$. Finally the center issues a smart card to user i after properly checking his physical identity. This smart card includes the set of integers $(n,\ g,\ e,\ c,\ S_i)$. If no new users are expected, the center can abort numbers p, q and d after all of the data is distributed. Hence, p, q, and d are kept secret from all users, S_i is known only to user i, and n, g, e, c are public and common to all the users.

2.2 Second phase

During the second phase of Type-1, the conference key is generated and simultaneously distributed among $m(\leq M)$ users. Users are connected in a ring so that user i always sends messages to user $i+1$ and user m sends to user 1. For simplicity, an expression of the user label is interpreted as modulo m so that it is between 1 and m. The key generation algorithm is the same for each user. Therefore, it is sufficient to describe the procedure for one user, labeled i, as follows:

step 1: User i generates a random number R_i, and keeps it secret. He sends $(X_i,\ Y_i,\ Z_i)$:

$$X_i = g^{eR_i} \bmod n, \tag{2.3}$$

$$Y_i = S_i g^{cR_i} \bmod n, \tag{2.4}$$

$$Z_i = 1, \tag{2.5}$$

to user $i+1$

step j $(2 \leq j \leq m-1)$: User i receives $(X_{i-1},\ Y_{i-1},\ Z_{i-1})$. He computes T_{i-1}:

$$T_{i-1} = X_{i-1} Z_{i-1}^{e} \bmod n. \tag{2.6}$$

He checks whether the following congruence holds:

$$(Y_{i-1}^{e} / T_{i-1}^{c})^{e^{M-j}} \equiv \prod_{1 \leq k \leq j-1} I_{i-k} \pmod{n}. \tag{2.7}$$

If this check succeeds, user i can verify that the message came via user $i-1$, user $i-2$, ..., and user $i-j+1$ successively. He then sends $(X_i,\ Y_i,\ Z_i)$:

$$X_i = X_{i-1}^{eR_i} \bmod n, \tag{2.8}$$

$$Y_i = Y_{i-1}^{e} S_i^{e^{j-1}} X_{i-1}^{cR_i} \bmod n, \tag{2.9}$$

$$Z_i = T_{i-1}, \tag{2.10}$$

to user $i+1$. Then he proceeds to the next step $j+1$.

step m: User i receives $(X_{i-1},\ Y_{i-1},\ Z_{i-1})$. He computes T_{i-1} by (2.6). He checks whether (2.7) for $j = m$ holds. If the check succeeds, user i can verify that the message came via user $i-1$, user $i-2$, ..., and user $i-m+1$ successively. He then computes conference key K:

$$K = X_{i-1}^{R_i} \bmod n. \tag{2.11}$$

The value of K is the same for all users, because

$$K = g^{e^{m-1} R_1 R_2 \ldots R_m} \bmod n. \tag{2.12}$$

2.3 Example of Type-1

Let $m = 4$ and $M = 10$. Transmitted messages $(X_1,\ Y_1,\ Z_1)$ from user 1 to user 2 at intervals of each step are as follows:

$$(g^{eR_1},\ S_1 g^{cR_1},\ 1), \qquad \text{after step 1;}$$

$$(g^{e^2 R_4 R_1},\ S_4^e g^{ceR_4} S_1^e g^{ceR_4 R_1},\ g^{eR_4}), \qquad \text{after step 2;}$$

$$(g^{e^3 R_3 R_4 R_1},\ S_3^{e^2} g^{ce^2 R_3} S_4^{e^2} g^{ce^2 R_3 R_4} S_1^{e^2} g^{ce^2 R_3 R_4 R_1},\ g^{e^2 R_3} g^{e^2 R_3 R_4}), \quad \text{after step 3.}$$

User 1 authenticates the messages if the following congruences hold at each step:

$$(Y_4^e/T_4^c)^{e^8} \equiv (S_4^e g^{ceR_4}/g^{ceR_4})^{e^8} \equiv S_4^{e^9} \equiv I_4 \pmod{n}, \qquad \text{at step 2;}$$

$$(Y_4^e/T_4^c)^{e^7} \equiv (S_3^{e^2} g^{ce^2 R_3} S_4^{e^2} g^{ce^2 R_3 R_4}/g^{ce^2 R_3} g^{ce^2 R_3 R_4})^{e^7}$$

$$\equiv S_3^{e^9} S_4^{e^9} \equiv I_3 I_4 \pmod{n}, \qquad \text{at step 3;}$$

$$(Y_4^e/T_4^c)^{e^6} \equiv (S_2^{e^3} g^{ce^3 R_2} S_3^{e^3} g^{ce^3 R_2 R_3} S_4^{e^3} g^{ce^3 R_2 R_3 R_4}/g^{ce^3 R_2} g^{ce^3 R_2 R_3} g^{ce^3 R_2 R_3 R_4})^{e^6}$$

$$\equiv S_2^{e^9} S_3^{e^9} S_4^{e^9} \equiv I_2 I_3 I_4 \pmod{n}, \qquad \text{at step 4.}$$

Finally he obtains conference key K:

$$K = X_4^{R_1} = g^{e^3 R_1 R_2 R_3 R_4}.$$

3. ID-based CKDS in a complete graph (Type-2)

3.1 First phase

During the first phase of Type-2, 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:

$$ed \equiv 1 \pmod{L}, \qquad L = \text{lcm}\ ((p-1),\ (q-1),\ (r-1)), \qquad 3 \leq e,\ d < L, \qquad (3.1)$$

where e is coprime to L. It also determines a prime c ($3 \leq$ c $< L$), and an integer g which is a primitive element over $GF(p)$, $GF(q)$, and $GF(r)$. For user i whose identification information is I_i, the center calculates integer S_i:

$$S_i = I_i^d \bmod nr. \qquad (3.2)$$

Note that $I_i = S_i^e \bmod nr$. Finally the center gives the set of integers $(n,\ r,\ g,\ e,\ c,\ S_i)$ to user i in a way similar to that of Type-1. If no new users are expected, the center can abort numbers

p, q and d after all of the data is distributed. Hence, p, q, and d are kept secret from all users, S_i is known only to user i, and n, r, g, e, c are public and common to all the users.

3.2 Second phase

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:

step 1: User j generates a random number U_j, and sends E_j:

$$E_j = g^{eU_j} \bmod n \tag{3.3}$$

to user i. He keeps U_j secret.

step 2: User i generates a random number P_i that is coprime to $(r-1)$. He computes $\overline{P}_i$:

$$P_i\overline{P}_i \equiv 1 \pmod{(r-1)}, \tag{3.4}$$

and keeps P_i and $\overline{P}_i$ secret. He generates a random number V_i and keeps it secret. He then sends $(X_i,\ Y_i,\ Z_{ij},\ F_i)$:

$$X_i = g^{eP_i} \bmod nr, \tag{3.5}$$

$$Y_i = S_i g^{cP_i} \bmod nr, \tag{3.6}$$

$$Z_{ij} = E_j^{P_i} \bmod n, \tag{3.7}$$

$$F_i = X_i^{eV_i} \bmod n, \tag{3.8}$$

to user j.

step 3: User j receives $(X_i,\ Y_i,\ Z_{ij},\ F_i)$. He checks whether the following $2(m-1)$ congruences hold:

$$Y_i^e / X_i^c \equiv I_i \pmod{nr}, \tag{3.9}$$

$$Z_{ij} \equiv X_i^{U_j} \pmod{n}. \tag{3.10}$$

If (3.9) and (3.10) hold, user j can verify that the message came from user i. User j generates a secret random number R_j. He then sends $(A_{ji},\ B_{ji},\ C_{ji})$:

$$A_{ji} = X_i^{eR_j} \bmod nr, \tag{3.11}$$

$$B_{ji} = S_j X_i^{cR_j} \bmod nr, \tag{3.12}$$

$$C_{ji} = F_i^{R_j} \bmod n, \tag{3.13}$$

to user i. He keeps $A_{jj} = X_j^{eR_j}$.

step 4: User i receives $(A_{ji},\ B_{ji},\ C_{ji})$. He checks whether the following $2(m-1)$ congruences hold:

$$B_{ji}^e / A_{ji}^{\mathrm{c}} \equiv I_j \pmod{nr}, \tag{3.14}$$

$$C_{ji} \equiv A_{ji}^{V_i} \pmod{n}. \tag{3.15}$$

If (3.14) and (3.15) hold, user i can verify that the message came from user j. He then computes conference key K:

$$K = (\prod_{j=1}^{m} A_{ji})^{\overline{P}_i} \bmod r. \tag{3.16}$$

The value of K is the same for all users, because

$$K = g^{e^2(P_i R_1 + P_i R_2 + \ldots + P_i R_m)\overline{P}_i} \bmod r = g^{e^2(R_1 + R_2 + \ldots + R_m)} \bmod r. \tag{3.17}$$

4. ID-based CKDS in a star (Type-3)

Type-2 can be simplified by restricting the process so that $j = 1$ and $2 \leq i \leq m$. Therefore, users are connected in a star network so that messages are transmitted between user 1 and user i $(2 \leq i \leq m)$. 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 among m users.

The key generation algorithm during the second phase of Type-3 is similar to that of Type-2. Note that user 1 can compute conference key $K = g^{e^2 R_1}$ at any time. User i $(2 \leq i \leq m)$ computes conference key K at step 4 by:

$$K = A_{1i}^{\overline{P}_i} \bmod r. \tag{4.1}$$

The value of K is the same for all users, because

$$K = g^{e^2 R_1 P_i \overline{P}_i} \bmod r = g^{e^2 R_1} \bmod r. \tag{4.2}$$

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

5. Security

The security of the proposed systems is based on the difficulty of deriving secret keys such as (p, q, d, S_i, R_i, K) in Type-1 and (p, q, d, S_i, U_i, V_i, P_i, $\overline{P}_i$, R_i, K) in Type-2 and Type-3 from public keys, transmitted messages, and other user's secret keys.

(1) Secrecy of (p, q, d, S_i) in all ICKDSs is based on the difficulty of factoring a large number n. If an opponent were able to factor n, he could then compute d from e and L, and could then obtain S_i from d and I_i.

(2) Secrecy of (R_i, K) in Type-1 and (U_i, V_i) in Type-2 and Type-3 is based on the difficulty of both factoring a large number n and computing discrete logarithms over $GF(p)$ and $GF(q)$. If an opponent were able to factor $n = pq$, he could then compute $g^{eR_i} \bmod p$ and $g^{eR_i} \bmod q$ in Type-1. Moreover, if he were able to compute discrete logarithms over $GF(p)$ and $GF(q)$, he could then compute R_i from $g^{eR_i} \bmod p$ and $g^{eR_i} \bmod q$, and could then obtain K from R_i in Type-1. Similarly, he could compute U_i and V_i, and could obtain Z_{ji} and C_{ji} for passing the check of (3.10) and (3.15), respectively, in Type-2 and Type-3.

(3) Secrecy of (P_i, $\overline{P}_i$, R_i, K) in Type-2 and Type-3 is based on the difficulty of computing discrete logarithm over $GF(r)$. Since an opponent can compute $\overline{e}$ such that $e\overline{e} \equiv 1 \pmod{(r-1)}$, he can easily derive $g^{P_i} \bmod r$ from X_i. If an opponent were able to derive $P_i \bmod (r-1)$ from $g^{P_i} \bmod r$, he could then compute $\overline{P}_i$ and could obtain K from $\overline{P}_i$ and A_{ji}.

(4) The best known algorithm for factoring $n = pq$ ($p < q$) requires a running time of $O(\exp((2 + o(1))\sqrt{\log p \log \log p}))$ [9]. Therefore, the designer can choose the sizes of p and q so as to prevent an impersonation attack in all ICKDSs and to ensure the secrecy of K in Type-1. The best known algorithm for computing the discrete logarithm over $GF(r)$ for any prime r requires a running time of $O(\exp((1 + o(1))\sqrt{\log r \log \log r}))$ [10]. Therefore, the designer can choose the size of r to ensure the secrecy of K in Type-2 and Type-3. 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.

(5) The Type-1 scheme corresponds to the most secure version of Ingemarsson's schemes [5] because K has an exponent degree m in an indeterminate R_i.

(6) If an opponent changes transmitted messages (X_i, Y_i) to ($X_i a^e$, $Y_i a^c$) in all ICKDSs, or (A_{ji}, B_{ji}) to ($A_{ji} b^e$, $B_{ji} b^c$) in Type-2 and Type-3, he can disturb the key distribution system by bypassing the ID check. As a result of this disturbance, each user obtains a different

conference key. However, user i and user j $(1 \leq i,\ j \leq m,\ i \neq j)$ can verify the sameness of their keys K by testing that an encrypted message with one's key is successfully decrypted with other's key, called an encryption-and-decryption test. Therefore, this disturbance is detectable.

If the Shamir-Fiat identity-based signature schemes [1, 2] are used to send messages, the disturbance during the transmission can be detected directly after the transmission. The Shamir-Fiat schemes realize both sender authentication and message authentication, while our proposed schemes realize only sender authentication. In key distribution systems, message authentication can be realized after an encryption-and-decryption test.

In the Shamir-Fiat schemes, user i has to know I_j exactly because it is used as input data in the verification process. In our proposed schemes, even if user i remembers I_j imperfectly, sender authentication is possible because he only checks whether he obtains a reasonable I_j as the output data in the verification process.

(7) In Type-1, user $i-1$ can derive S_i^e mod n from his secret key R_{i-1} and transmitted messages $X_i = g^{eR_i}$ after step 1 and $Y_i = S_{i-1}^e g^{ceR_{i-1}} S_i^e g^{ceR_{i-1}R_i}$ after step 2. Even if user $i-1$ obtains S_i^e mod n, it is difficult to derive S_i mod n from S_i^e mod n. Therefore, user $i-1$ cannot pretend to be user i.

(8) The checks of (3.10) and (3.15) in Type-2 and Type-3 are introduced to detect impersonation attacks using passive and active wiretaps. Since the purpose and function of (3.10) and (3.15) is the same, we describe the case for (3.10) as an example.

If the check of (3.10) and related computations of (3.3) and (3.7) are omitted, an opponent can pretend to be user i and finally obtains K as follows: After the opponent wiretaps (X_i mod nr, Y_i mod nr), he can produce the following ($\widetilde{X}_i$ mod nr, $\widetilde{Y}_i$ mod nr) to pass the check of (3.9):

$$P'\overline{P'} \equiv 1 \pmod{(r-1)}, \qquad e\overline{e} \equiv 1 \pmod{(r-1)}, \tag{5.1}$$

$$X_i' = g^{eP'} \bmod r, \qquad Y_i' = (I_i(X_i')^c)^{\overline{e}} \bmod r, \tag{5.2}$$

$$\begin{cases} \widetilde{X}_i \equiv X_i \bmod n, \\ \widetilde{X}_i \equiv X_i' \bmod r, \end{cases} \qquad \begin{cases} \widetilde{Y}_i \equiv Y_i \bmod n, \\ \widetilde{Y}_i \equiv Y_i' \bmod r, \end{cases} \tag{5.3}$$

He sends $(\widetilde{X}_i,\ \widetilde{Y}_i)$ to user j and gets $\widetilde{A}_{ji} = \widetilde{X}_i^{eR_j}$ from user j. From the unfairly generated $\widetilde{A}_{ji}$ and $\overline{P'}$, he computes $\widetilde{A}_{ji}^{\overline{P'}} = g^{e^2 R_j}$ and finally obtains K. Note that this impersonation attack is made without the knowledge of S_i mod nr.

If the check of (3.10) is introduced, an opponent cannot pretend to be user i who is absent from a current ICKDS. The opponent generates $(\widetilde{X}_i,\ \widetilde{Y}_i)$ from X_i mod nr previously generated by regular user i. However, he cannot produce Z_{ij} to pass the check of (3.10) because he does not know the previous value of P_i that is a parameter of X_i and Z_{ij}. Note that the previously generated Z_{ij} is not reusable because the value of U_j changes at each ICKDS. Therefore, the check of (3.10) detects this impersonation attack.

As pointed out in [7], if an opponent tries to pretend to be user i who is present at the current ICKDS, the check of (3.10) is ineffective. The opponent interrupts transmission from user i, and sends $(\widetilde{X}_i,\ \widetilde{Y}_i,\ Z_{ij})$ generated from a current $(X_i,\ Y_i,\ Z_{ij})$. Since this Z_{ij} passes the check of (3.10), the opponent can get correct key K. However, regular user i receives $\widetilde{A}_{ji}$ generated from the false P', and he finally obtains false key $\widetilde{K} = g^{e^2(P' \sum_{j=1}^{m} R_j)\overline{P}_i} \bmod r$. In this one-directional attack, the opponent cannot get this false key $\widetilde{K}$. Therefore, the one-directional real time impersonation attack is detectable after the encryption-and-decryption test by regular user i [7].

By extending the above analysis, Yacobi [11] showed a bidirectional real time attack between user i and user j in a star system (Type-3). In his attack using false random number R'_j, the opponent can get the same false key $\widetilde{K}' = g^{e^2 R'_j} \bmod r$ as user i gets. His attack method can be generalized to a complete-graph system (Type-2). Since the opponent can hold both a correct key and a false key, this bidirectional impersonation attack would be successful if the opponent could change all interactions with regular user i after the key generation. If a conference is carried out in radio broadcast networks, this attack would be detectable by the encryption-and-decryption test because it seems to be physically impossible for the opponent to change the radio transmissions.

6. Transmission efficiency

Type-1 requires $(m-1)$ steps for transmission because messages must be sent sequentially among m users. However, Type-2 and Type-3 requires 3 steps for transmissions for any m because messages can be broadcasted simultaneously. The total numbers of message transmissions among m users in Type-1, Type-2, and Type-3 are $m(m-1)$, $3m(m-1)$, and $3(m-1)$, respectively. The total message length of transmissions in each Type-1, Type-2, and Type-3 is $3m(m-1) \log n$, $4m(m-1)(\log nr + \log n)$ and $4(m-1)(\log nr + \log n)$ bits, respectively. We compare the above indices for transmission efficiency among each type. For the numbers of sequential steps, Type-2 and Type-3 are better than Type-1 if $m \geq 5$. For the total numbers of message transmissions, Type-3 is better than Type-1 if $m \geq 4$. For the total message length of transmissions, assuming $\log n$=512 and $\log r$=256, Type-3 is better than Type-1 if $m \geq 4$. Summarizing the comparisons from the viewpoint of transmission efficiency, Type-1 is the best if $m = 3$, and Type-3 is the best if

$m \geq 4$. Note that if m=2, then it is the best to use the simpler schemes such as [1, 2, 3]. Expansion of message transmission is needed to ensure the security of a conference key.

Acknowledgement: We would like to thank T. Okamoto of NTT Corporation for his valuable discussions.

References

[1] SHAMIR, A. :'Identity-based cryptosystems and signature schemes', Proceedings of Crypto-84, Santa Barbara, August 1984, pp47-53

[2] FIAT, A. and SHAMIR, A. :'How to prove yourself: Practical solutions to identification and signature problems', Proceedings of Crypto-86, Santa Barbara, August 1986, pp18-1–18-7

[3] OKAMOTO, E.:'Proposal for identity-based key distribution systems', *Electron. Lett.*, 1986, **22**, pp1283-1284

[4] DIFFIE, W., and HELLMAN, M. E. :'New directions in cryptography', *IEEE Trans.* 1976, **IT-22**, pp644-654

[5] INGEMARSSON, I, TANG, D. T. and WONG, C. K. :'A conference key distribution system', *IEEE Trans.* 1982, **IT-28**, pp714-720

[6] KOYAMA, K. :'Identity-based conference key distribution system', *IEE Electron. Lett.*, May 7, 1987, Vol.23, No.10, pp.495-496.

[7] KOYAMA, K. and OHTA, K. :'Identity-based conference key distribution systems in broadcast networks' *IEE Electron. Lett.*, June 4, 1987, Vol.23, No.12, pp.647-649.

[8] RIVEST, R. L., SHAMIR, A., and ADLEMAN, L.:'A method for obtaining digital signatures and public-key cryptosystems', *Commun. ACM,* 1978, **21**, pp120-126

[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**, pp1-15

[11] YACOBI, Y. :'Attack on the Koyama-Ohta identity-based key distribution STAR system', private communication, July 13, 1987.