PAPER Special Section on Cryptography and Information Security

Optimal Unconditionally Secure ID-Based Key Distribution Scheme for Large-Scaled Networks[*]

Goichiro HANAOKA[†], *Nonmember*, Tsuyoshi NISHIOKA[††], Yuliang ZHENG[†††], *and* Hideki IMAI[†], *Regular Members*

SUMMARY Efficient ID-based key sharing schemes are desired worldwide in order to obtain secure communications on the Internet and other related networks, and Key Pre-distribution System (KPS) is one of the majority of such key sharing schemes. The remarkable property of KPS, is that, user need only input the partner's identifier to the secret KPS-algorithm in order to share a key between them. Although this is just a small part of many advantages KPS has in terms of efficiency, an enormous amount of memory is always required to achieve perfect security. While the conventional KPS methods can establish communication links between any pair of entities in a communication system, in most of the practical communication environment, such as in a broadcast system, not all links will be required. In this article, we achieved a desirable method to remove the unnecessary communication links between any pair of entities in a communication system. In our scheme, required memory size per entity was just proportional to the number of entities of the partner's, while that in conventional KPS, it is proportional to the number of entities of the whole communication system. As an example, if an entity communicates with only $1/r$ others, the memory requirement is reduced to $1/r$ of the conventional KPS's. Furthermore, it was proven that the obtained memory size was optimum. Overall, our scheme confirmed greater efficiency to achieve secure communication particularly suited in large-scale networks.

key words: *key predistribution system, ID-based cryptosystem, collusion attack*

1. Introduction

For information security, ID-based key distribution technologies are very important. The concept of ID-based key cryptosystems was originally proposed by Shamir [3], [4]. Maurer and Yacobi presents an ID-based key distribution scheme following Shamir's concept [5], [6]. However, their scheme requires a huge computational power. Okamoto and Tanaka [7] also proposed a key-distribution scheme based on a user's identifier, but it requires prior communications between a sender and a receiver to share the employed key. Thus, the performance of these schemes is unsatisfactory. However, Blom's ID-based key-distribution scheme [2], which is generalized by Matsumoto and Imai [1], has very good properties in terms of computational complexity and non-interactivity. Many useful schemes based on Blom's scheme have been proposed [1], [11]–[17], and they are called Key Predistribution Systems (KPS) [1].

In a KPS, no previous communication is required and its key-distribution procedure consists of simple calculations. Furthermore in order to share the key, a participant should only input its communication partner's identifier to its secret KPS-algorithm. Blundo et al. [14], [15], Kurosawa et al. [18], [19] showed lower bounds on memory size of users' secret algorithms and developed KPS for a conference-key distribution. Moreover Fiat and Naor [16], Kurosawa et al. [19] applied a KPS for a broadcasting encryption system.

Although KPS has many desired properties, the following problem exists: When a number of users, which exceeds a certain threshold, they can calculate the central authority's secret information. Thus, to achieve perfect security the collusion threshold is determined to be larger than the number of entities in the network. Setting up such a high collusion threshold in this scheme requires large amounts of memory in the center as well as for the users. Solving this problem will make KPS much more attractive for ID-based key-distribution. Tsujii and others [9] made attempts to enhance the KPS's security level with less amounts of memory. However, it has been shown that majority of their atttempted schemes are of equivalent level as of the conventional KPS [10], and moreover, their other schemes seem to be too complex to prove their accurate security.

Although KPS provides common keys for all possible communication links among entities, in practical communication systems most of them are not necessary. By removing such unnecessary communication links, we can significantly reduce the required memory[**]. In our

Manuscript received March 24, 2000.

Manuscript revised July 12, 2000.

[†] The authors are with the 3rd Department, Institute of Industrial Science, The University of Tokyo, Tokyo, 106-8558 Japan. The first author is supported by a Research Fellowship from Japan Society for the Promotion of Science (JSPS).

[††] The author is with the Information Technology R&D Center, Mitsubishi Electric Corporation, Kamakura-shi, 247-8501 Japan.

[†††] The author is with the Peninsula School of Computing and Information Technology Monash University, Melbourne, VIC 3199, Australia.

[*] A part of this work was presented at The Second International Conference on Information and Communication Security (ICICS'99) [22]. A part of this work was performed in part of Research for the Future Program (RFTF) supported by Japan Society for the Promotion of Science (JSPS) under contract no. JSPS-RFTF 96P00604.

scheme, the required memory for each entity is proportional to the number of its communication partners, while in the conventional KPS it is proportional to the number of entities in the whole system. E.g., if an entity communicates only with $1/r$ of others, the required memory is reduced for a factor $1/r$ in comparison with the conventional KPS. Furthermore, this memory size is proven optimal. In this work, we also propose an optimal *asymmetric t-conference key distribution scheme*. Since this scheme has good properties, it is considered to be utilized effectively in other applications.

Section 2 gives a brief review of the KPS. Then, in Sect. 3, straight-forward implementation of our scheme and the corresponding problem are described. Section 4 explains an asymmetric t-conference key scheme as the solution of straight-forward implementation problem. Afterwards, an optimal construction based on asymmetric t-conference key distribution scheme is described in Sect. 5. This is followed by the evaluation and discussion of the security of our scheme in Sect. 6. In Sect. 7, we show a modification of our scheme. Finally, Sect. 8 closes the paper with some concluding remarks.

2. A Brief Overview of KPS

A KPS consists of two kinds of entities: One entity is the KPS center, the other entities are the users who want to share a common key. The KPS center possesses the KPS-center algorithm by which it can generate an individual secret algorithm for each user. These individual algorithms are (pre-) distributed by the center to their users and allow each user to calculate a common key from the ID of his communication partner. This section explains how the users' secret KPS-algorithms are generated and how users share a common key.

Let a symmetric function $G(x,y)$ be the KPS-center algorithm. Then, each entity u_i $(i=1,2,\cdots,N)$ is given the secret algorithm $U_{u_i}(x)(=G(x,u_i))$ $(i=1,2,\cdots,N)$, respectively. In order to share the communication key between u_i and u_j, they should simply input u_j and u_i to their secret algorithms, respectively. Since $G(x,y)$ is a symmetric function, they both obtain $k_{u_iu_j}=U_{u_i}(u_j)=U_{u_j}(u_i)=G(u_i,u_j)$.

KPS has three noteworthy properties. First, there is no need to send messages for the key distribution between entities who want to establish a cryptographic communication channel. Second, its key-distribution procedure consists of simple calculations so that its computational costs are quite small. Finally, in order to share the key, a participant has only to input its communication partner's identifier to its secret algorithm. Thus, KPS is well applicable to one-pass or quick-response transactions, e.g. mail systems, broadcasting systems, electronic toll collection systems, and so on.

However, KPS has a certain collusion threshold; when more users cooperate they can calculate the KPS-center algorithm $G(x,y)$. Thus, to achieve perfect security the collusion threshold should be determined to be larger than the number of entities in the network. Accordingly, the required memory for users' secret algorithms are increased due to the collusion threshold. In the following subsection, the relationship between the memory size and the collusion threshold is discussed in more detail.

2.1 A Lower Bound of $|U_{u_i}(x)|$

For a random variable X, $H(X)$ denotes the entropy of X. Generally,

$$0\le H(X)\le \log_2|X|, \tag{1}$$

where $X=\{x\mid \Pr(X=x)>0\}$. In particular, $H(X)=\log_2|X|$ iff X is uniformly distributed.

Blundo et al. [14] showed a lower bound of required memory size for users as follows:

Proposition 1 ([14]): Suppose that for each $\mathcal{P}\subseteq\{u_1, u_2,\cdots,u_N\}$ such that $|\mathcal{P}|=t$, there is a key $k_{\mathcal{P}}$ associated with $\mathcal{P}$. Each user $u_i\in\mathcal{P}$ can compute $k_{\mathcal{P}}$, and if colluders $\mathcal{F}\subseteq\{u_1,u_2,\cdots,u_N\}$, $|\mathcal{F}|\le\omega$ and $|\mathcal{F}\cap\mathcal{P}|=0$, where ω is the collusion threshold in the system, then $\mathcal{F}$ cannot obtain any information about $k_{\mathcal{P}}$. Accordingly, a lower bound of the required memory size for users' secret algorithm U_{u_i} is estimated as follows:

$$\log_2|U_{u_i}|\ge\left(\begin{array}{c}t+\omega-1\\ t-1\end{array}\right)H(K), \tag{2}$$

where K is a random variable which takes on the key space $\mathcal{K}$, and $k_{\mathcal{P}}\in\mathcal{K}$ for any $\mathcal{P}$. Note that the key length of a shared key is $H(K)$. In order to achieve perfect security, $\omega+t$ must be equal to the number of entities in the whole network. For $t=2$, Eq. (2) becomes $\log_2|U_{u_i}|\ge(\omega+1)H(K)$.

2.2 Optimal Schemes

Blundo et al. [14] presented a KPS which achieves the optimal memory size. In this scheme, the center chooses a random symmetric polynomial in t variables over $GF(q)$ in which the degree of any variable is at most ω, that is, a polynomial

$$f(x_1,\cdots,x_t)=\sum_{i_1=0}^{\omega}\cdots\sum_{i_t=0}^{\omega}a_{i_1\cdots i_t}{x_1}^{i_1}\cdots{x_t}^{i_t}, \tag{3}$$

where $a_{i_1\cdots i_t}=a_{\sigma(i_1\cdots i_t)}$ for any permutation

[**] We have shown implementations of this concept for broadcasting and electronic toll collection systems (ETC) in [17] and [20], respectively. Our new scheme in this paper can be regarded as a generalized version of the above schemes.

σ on $(i_1, \cdots, i_t)$. The center computes $U_{u_i} = f(x_1, x_2, \cdots, x_t)|_{x_1=u_i}$ and gives U_{u_i} $(i = 1, \cdots, N)$ to u_i $(i = 1, \cdots, N)$, respectively. Then, they can share their communication keys by inputting $t-1$ partners' identifiers. For $t = 2$, Blom's scheme [2], Matsumoto-Imai scheme [1] and some others are also known as being optimal schemes. Although these schemes achieve the optimal memory size: $\log_2 |U_{u_i}| = (\omega + 1)H(K)$ (See Proposition 1), the amount of memory is still large (For perfect security, ω must be equal to $N-2$). Especially, in large-scale networks required memory size is enlarged according to the high collusion threshold. Furthermore, on a smart card since its size of storage is strictly limited, the collusion threshold cannot be set up high enough to avoid strong collusion attacks by huge number of entities. For example, $\omega = 8191$ is selected as the collusion threshold in "KPSL1 card" [21], where the key length is 64bits. The secret-algorithm itself then consumes 64-KBytes of memory size in each IC card. Therefore KPS was considered to be somewhat expensive for real IC card systems at that time. By introducing 128–256 bits symmetric key cryptosystems (namely, $H(K) = 128$–256 bits), this problem will be more serious.

3. Straight-Forward Method for Removing Unnecessary Functions

As already mentioned, some KPSs are proven to be optimal, and it is impossible to reduce the required memory size providing all the communication links. However, the required memory size of these schemes are still high. Although a possibility to reduce the required memory is to reduce the collusion threshold or the key length, the security is also reduced considerably. So, we pay attention to the unnecessary communication links in networks in order to reduce the memory size maintaining the same security level. In this section, we show a basic concept to attain the specified goal and we consider its requirements.

3.1 Unnecessary Communication Links

In a large-scale network, there are many pairs of entities that do not communicate with each other at all. As an illustration we point out the following two reasons. Firstly, to avoid the illegal access, certain users are not allowed to access specific resources by access controlling techniques. Secondly, there exist many resources which perform only to some specific client computers. Also, pairs of entities, which are not related with each other, do not need to communicate with each other.

Although there are many unnecessary communication links, conventional KPSs cannot deal with them efficiently. Namely, in conventional KPSs it seems to be impossible to remove only unnecessary communication links.

3.2 Straight-Forward Implementation and Its Problem

A possible solution to reduce memory size by removing unnecessary communication links is to construct a whole large network by using small KPSs. Namely, if small KPSs are provided only for necessary communication links, then the unnecessary ones can be removed. However, straight-forward implementation of this approach has a serious problem. The required memory can be even more than that of the conventional KPS. We explain this problem in more details below.

Here, we assume that the set of all entities $\mathcal{U}$ are devided to $N_{\mathcal{U}}$ subsets $\{\mathcal{U}_1, \mathcal{U}_2, \cdots, \mathcal{U}_{N_{\mathcal{U}}}\}$ $(|\mathcal{U}_i \cap \mathcal{U}_j| = 0)$. Each $\mathcal{U}_i$ $(i = 1, \cdots, N_{\mathcal{U}})$ fulfills the following condition: The set of all communication partners of all $u_i \in \mathcal{U}_i$ is $\hat{\mathcal{U}}_i = \{\mathcal{U}_{i,1}, \mathcal{U}_{i,2}, \cdots, \mathcal{U}_{i,N_i}\}$, where $\exists j$ $\mathcal{U}_{i,k} = \mathcal{U}_j$ $(k = 1, 2, \cdots, N_i)$ and $N_i \le N_{\mathcal{U}}$, and for $i_1, i_2 \in \{1, 2, \cdots, N_{\mathcal{U}}\}$, if $\mathcal{U}_{i_1} \in \hat{\mathcal{U}}_{i_2}$, then $\mathcal{U}_{i_2} \in \hat{\mathcal{U}}_{i_1}$. For convenience, if $\mathcal{U}_{i_1} \in \hat{\mathcal{U}}_{i_2}$, we say $< i_1, i_2 >= 1$, otherwise, $< i_1, i_2 >= 0$. In this paper, we assume that each entity's identity includes the infomation of the subset which the entity belongs to[†].

Then, if a whole network is constructed by small KPSs straight-forwardly, the critical problems will appear in following two situations:

Case1: $< i_1, i_2 >= 1$, $< i_1, i_1 >= 0$ $(i_1 \ne i_2)$

For the communication $< i_1, i_2 >= 1$, a small KPS is provided. The collusion threshold of this KPS is determined to be equal to $|\mathcal{U}_{i_1}| + |\mathcal{U}_{i_2}| - 2$. From Eq. (2), the required memory size for this communication is also proportional to $|\mathcal{U}_{i_1}| + |\mathcal{U}_{i_2}| - 1$. However, since $< i_1, i_1 >= 0$, even $u_{i_1} \in \mathcal{U}_{i_1}$ communicates only with $|\mathcal{U}_{i_2}|$ entities using the above memory.

Case2: $< i_1, i_2 >= 1$, $< i_1, i_3 >= 1$, $< i_2, i_3 >= 0$ $(i_1 \ne i_2,\ i_1 \ne i_3,\ i_2 \ne i_3)$

For these communications, we consider two kinds of construction of small KPSs: a KPS for $\{\mathcal{U}_{i_1}, \mathcal{U}_{i_2}\}$ and a KPS for $\{\mathcal{U}_{i_1}, \mathcal{U}_{i_3}\}$ are set up, or a KPS for $\{\mathcal{U}_{i_1}, \mathcal{U}_{i_2}, \mathcal{U}_{i_3}\}$ is set up. For the first construction, the collusion threshold of two KPSs are $|\mathcal{U}_{i_1}| + |\mathcal{U}_{i_2}| - 2$ and $|\mathcal{U}_{i_1}| + |\mathcal{U}_{i_3}| - 2$, respectively. Thus, the required memory size of $u_{i_1} \in \mathcal{U}_{i_1}$ for these communications is proportional to $2|\mathcal{U}_{i_1}| + |\mathcal{U}_{i_2}| + |\mathcal{U}_{i_3}| - 2$. On the other hand, for the second construction the collusion threshold of the KPS is $|\mathcal{U}_{i_1}| + |\mathcal{U}_{i_2}| + |\mathcal{U}_{i_3}| - 2$. Hence,

[†] For example, if an entity is a terminal of a (local-area) network, the information of the subset which the entity belongs to can be extracted from the network domain name or IP address of the entity, assuming that the terminals in the network are equally access controlled to other external networks.

the amount of memory of $u_{i_1} \in \mathcal{U}_{i_1}$ is proportional to $|\mathcal{U}_{i_1}|+|\mathcal{U}_{i_2}|+|\mathcal{U}_{i_3}|-1$. Further, the required memory size for $u_{i_2} \in \mathcal{U}_{i_2}$ and $u_{i_3} \in \mathcal{U}_{i_3}$ are also proportional to $|\mathcal{U}_{i_1}|+|\mathcal{U}_{i_2}|+|\mathcal{U}_{i_3}|-1$. Anyway, in both constructions since $u_{i_1} \in \mathcal{U}_{i_1}$ communicate only with $|\mathcal{U}_{i_2}|+|\mathcal{U}_{i_3}|$ entities by these KPS(s), the required memory size for u_{i_1} is large comparing with the amount of memory. Moreover, in the second construction $u_{i_2} \in \mathcal{U}_{i_2}$ and $u_{i_3} \in \mathcal{U}_{i_3}$ communicate only with $|\mathcal{U}_{i_1}|$ entities. Hence, their required memory size is also large.

In the worst case, the required memory size for an entity is almost 2 times of that in conventional KPS. Namely, if $< i_1, i >= 1$ ($i = 1, \cdots, N_{\mathcal{U}},\ i \neq i_1$) and $< j, k >= 0$ ($j,\ k = 1, \cdots, N_{\mathcal{U}},\ j,\ k \neq i_1$), then, the required memory size is $\left(\sum_{i\in\{1,2,\cdots,N_{\mathcal{U}}\}, i\neq i_1} (|\mathcal{U}_i| + |\mathcal{U}_{i_1}| - 1)\right) H(K)$. On the other hand, in conventional KPS the required memory size is $\left(\sum_{i\in\{1,2,\cdots,N_{\mathcal{U}}\}, i\neq i_1} |\mathcal{U}_i| - 1\right) H(K)$. Hence, straight-forward implementation of constructing a whole network by small KPSs is inefficient. Note that **Case1** and **Case2** can not be considered as rare. These cases cannot occur iff $\mathcal{U}$ can be divided to hold the following condition:

$$< i, j >= \begin{cases} 0 & (i \neq j) \\ 1 & (i = j) \end{cases} \qquad \forall i, \forall j \in \{1, 2, \cdots, N_{\mathcal{U}}\}.$$

4. Optimal Key Sharing Schemes

As mentioned in the previous section, we can not construct a large-scale network efficiently by only using normal KPSs. In this section, we show a new concept of key sharing schemes to construct a large-scale network optimally. The following lemma indicates the required key sharing schemes for optimal constructions in terms of memory size:

Lemma 1: The required memory size for entities is optimal if a whole network is constructed by optimal normal KPSs [1], [2], [14] for $< i, i >= 1$ for all $i \in \{1, \cdots, N_{\mathcal{U}}\}$ and other key sharing schemes whose memory size for $< i, j >= 1$ for all $i, j \in \{1, \cdots, N_{\mathcal{U}}\}$, $i \neq j$ is optimal for $< i, j >= 1$ for all $i, j \in \{1, \cdots, N_{\mathcal{U}}\}$, $i \neq j$.

Proof. In large scale networks, there are two kinds of communication, i.e. communication among the same subset and that between different two subsets. Therefore, if optimal key sharing systems for $< i, i >= 1$ for all $i \in \{1, \cdots, N_{\mathcal{U}}\}$ and those for $< i, j >= 1$ for all $i, j \in \{1, \cdots, N_{\mathcal{U}}\}$, $i \neq j$ are provided, a key sharing system for a large scale network can be constructed optimally.

Since optimal normal KPSs provide optimal memory size for $< i, i >= 1$ for all $i \in \{1, \cdots, N_{\mathcal{U}}\}$, the required memory size for entities becomes optimal if a whole network is constructed by optimal normal KPSs [1], [2], [14] for $< i, i >= 1$ for all $i \in \{1, \cdots, N_{\mathcal{U}}\}$ and other key sharing schemes whose memory size for $< i, j >= 1$ for all $i, j \in \{1, \cdots, N_{\mathcal{U}}\}$, $i \neq j$ is optimal for $< i, j >= 1$ for all $i, j \in \{1, \cdots, N_{\mathcal{U}}\}$, $i \neq j$. □

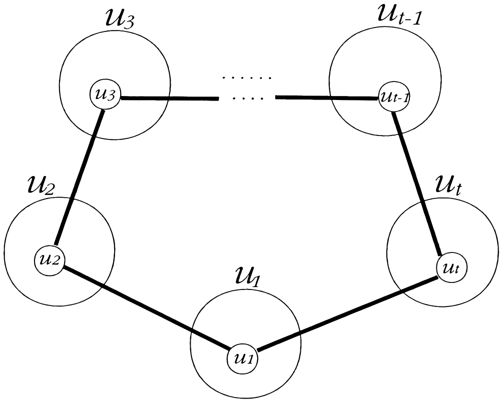


Fig. 1 An asymmetric t-conference key distribution.

4.1 Requirement for the Optimal Key Sharing Scheme

In this subsection, a lower bound of the memory size of a key sharing scheme for $< i_1, i_2 >= 1$ ($i_1 \neq i_2$) is shown. For other applications, we further generalize the key sharing scheme and call it *asymmetric t-conference key distribution*. Asymmetric t-conference key distribution is defined as follows:

Definition 1: Let $\mathcal{U}$ be a set of entities and $\mathcal{U}$ is divided to t subsets $\mathcal{U} = \{\mathcal{U}_1, \mathcal{U}_2, \cdots, \mathcal{U}_t\}$. A key-sharing scheme for $\mathcal{U}$ is called asymmetric t-conference key distribution if

1. $u_1 \in \mathcal{U}_1,\ u_2 \in \mathcal{U}_2,\ \cdots,\ u_t \in \mathcal{U}_t$ can compute their common key among them non-interactively (note that common keys among a same subset are not required).
2. Collusion thresholds $\psi_1, \psi_2, \cdots, \psi_t$ are independently set up for each of $\mathcal{U}_1, \mathcal{U}_2, \cdots, \mathcal{U}_t$. And unless a group of colluders $\mathcal{F}_i \in \mathcal{U}_i$ holds $|\mathcal{F}_i| > \psi_i$, any information of a common key is not exposed to entities who should not have it.

Figure 1 illustrates an example of an asymmetric t-conference key distribution. Note that a key sharing scheme for $< i_1, i_2 >= 1$ ($i_1 \neq i_2$) is an asymmetric 2-conference key distribution. Therefore, from Lemma 1 it is clear that optimal normal KPSs and optimal asymmetric 2-conference key distribution schemes are required to construct a optimal key distribution system for a large-scale network.

Lemma 2: In an asymmetric t-conference key distribution system, for $u_i \in \mathcal{U}_i$ the amount of memory of the secret algorithm U_i holds the following lower bound:

$$\log_2 |U_i| \geq \left(\prod_{j \in \{1,\cdots,t\}, j \neq i} (\psi_j + 1) \right) H(K). \tag{4}$$

Proof. The mutual information between random valuables X and Y fulfills following two equations:

$$I(X;Y) = H(X) - H(X|Y), \tag{5}$$

$$I(X;Y) = I(Y;X). \tag{6}$$

From Eq. (5) and Eq. (6), following equation is obtained:

$$H(X) = H(Y) - H(Y|X) + H(X|Y). \tag{7}$$

Let $\mathcal{U}_j{}'$ $(j = 1, \cdots, i-1, i+1 \cdots, t)$ be sets $\{u_j^0, \cdots, u_j^{\psi_j}\} \subseteq \mathcal{U}_j$ $(j = 1, \cdots, i-1, i+1 \cdots, t)$, respectively, and $\mathcal{K}_{u_i, u_1^{k_1}, \cdots, u_{i-1}^{k_{i-1}}, u_{i+1}^{k_{i+1}}, \cdots, u_t^{k_t}}$ be the set of all possible values of the shared key among $u_i, u_1^{k_1} \in \mathcal{U}_1', \cdots, u_{i-1}^{k_{i-1}} \in \mathcal{U}_{i-1}', u_{i+1}^{k_{i+1}} \in \mathcal{U}_{i+1}', \cdots, u_t^{k_t} \in \mathcal{U}_t'$ such that $k_j \in \{0, \cdots, \psi_j\}$, $j = 1, \cdots, i-1, i+1, \cdots, t$. Letting $\mathcal{K}_{u_i}$ be $\mathcal{K}_{u_i, u_1^0, \cdots, u_t^0} \times \cdots \times \mathcal{K}_{u_i, u_1^{\psi_1}, \cdots, u_t^{\psi_t}}$ ($\mathcal{K}_{u_i}$ is a Cartesian product of all $\mathcal{K}_{u_i, u_1^{k_1}, \cdots, u_{i-1}^{k_{i-1}}, u_{i+1}^{k_{i+1}}, \cdots, u_t^{k_t}}$ $(k_j = 0, \cdots, \psi_j,\ j = 1, \cdots, i-1, i+1 \cdots, t)$ and K_{u_i} be a random valuable which takes on $\mathcal{K}_{u_i}$, from Eq. (7), the entropy of U_i is described as follows:

$$H(U_i) = H(K_{u_i}) - H(K_{u_i}|U_i) + H(U_i|K_{u_i}). \tag{8}$$

Since in an asymmetric t-conference key distribution scheme, K_{u_i} can be computed by using U_i,

$$H(K_{u_i}|U_i) = 0. \tag{9}$$

Then, from Eq. (8) and Eq. (9) the following inequality is obtained:

$$H(U_i) = H(\mathcal{K}_{u_i}) + H(U_i|K_{u_i}) \geq H(K_{u_i}). \tag{10}$$

In a secure asymmetric t-conference key distribution system, no information of a shared key among a communication group can be obtained by other illegal entities unless the number of colluders in any $\mathcal{U}_j$ exceeds ψ_j. Therefore, letting $K_{u_i, u_1^{k_1}, \cdots, u_{i-1}^{k_{i-1}}, u_{i+1}^{k_{i+1}} \cdots, u_t^{k_t}}$ be a random valuable which takes on $\mathcal{K}_{u_i, u_1^{k_1}, \cdots, u_{i-1}^{k_{i-1}}, u_{i+1}^{k_{i+1}} \cdots, u_t^{k_t}}$, we obtain

$$\begin{aligned} & H(K_{u_i, {u_1}^{k_{1_0}}, \cdots, {u_{i-1}}^{k_{i-1_0}}, {u_{i+1}}^{k_{i+1_0}}, \cdots, {u_t}^{k_{t_0}}} \\ & \quad |K_{u_i, u_1^{k_1}, \cdots, {u_{i-1}}^{k_{i-1}}, {u_{i+1}}^{k_{i+1}}, \cdots, u_t^{k_t}} \\ & \qquad k_j = 0, \cdots, k_{j_0 - 1}, k_{j_0 + 1}, \cdots, \psi_j, \\ & \qquad j = 1, \cdots, i-1, i+1 \cdots, t) \\ & \quad = H(K_{u_i, u_1^{k_{1_0}}, \cdots, u_{i-1}^{k_{i-1_0}}, u_{i+1}^{k_{i+1_0}}, \cdots, u_t^{k_{t_0}}}) \\ & \quad = H(K), \end{aligned} \tag{11}$$

for all $k_{1_0}, \cdots, k_{t_0}$ $(k_{j_0} \in \{1, \cdots, \psi_j\},\ j = 1, \cdots, i-1, i+1, \cdots, t)$. Thus, for $H(K_{u_i})$ we have

$$H(K_{u_i}) = \left(\prod_{j \in \{1,\cdots,t\}, j \neq i} (\psi_j + 1) \right) H(K). \tag{12}$$

From Eq. (10) and Eq. (12), we obtain

$$H(U_i) \geq \left(\prod_{j \in \{1,\cdots,t\}, j \neq i} (\psi_j + 1) \right) H(K). \tag{13}$$

Hence, from Eq. (1) Eq. (13) becomes Eq. (4). □

Proposition 2: For perfect security, in an asymmetric t-conference key distribution system the amount of memory of the secret algorithm U_i holds following lower bound:

$$\log_2 |U_i| \geq \left(\prod_{j \in \{1,\cdots,t\}, j \neq i} |\mathcal{U}_j| \right) H(K). \tag{14}$$

Proof. For perfect security, each ψ_i $(i = 1, \cdots, t)$ must be equal to $|\mathcal{U}_i| - 1$. By introducing this collusion threshold, Eq. (4) becomes Eq. (14). □

4.2 An Example of Optimal Asymmetric t-Conference Key Distribution Schemes

In this subsection an optimal asymmetric t-conference key distribution scheme is shown. In this scheme, the symmetric polynomial which is the KPS-center algorithm in Blundo et al.'s scheme is replaced with an asymmetric polynomial in t variables $x_1, x_2, \cdots, x_t$ over $GF(q)$ in which the degree of each variable is $\psi_1, \psi_2, \cdots, \psi_t$, respectively, that is, a polynomial

$$f(x_1, \cdots, x_t) = \sum_{i_1=0}^{\psi_1} \cdots \sum_{i_t=0}^{\psi_t} a_{i_1 \cdots i_t} {x_1}^{i_1} \cdots {x_t}^{i_t}. \tag{15}$$

Note that $a_{i_1 \cdots i_t}$ is not necessary to be equal to $a_{\sigma(i_1 \cdots i_t)}$ for any permutation σ on $(i_1, \cdots, i_t)$. The center computes $U_i = f(x_1, x_2, \cdots, x_t)|_{x_i = u_i}$ and gives each U_i to u_i, respectively. When u_i communicates with $u_1 \in \mathcal{U}_1, u_2 \in \mathcal{U}_2, \cdots, u_{i-1} \in \mathcal{U}_{i-1}, u_{i+1} \in \mathcal{U}_{i+1}, \cdots, u_t \in \mathcal{U}_t$, u_i computes their communication keys by $U_i|_{x_1 = u_1, x_2 = u_2, \cdots, x_{i-1} = u_{i-1}, x_{i+1} = u_{i+1}, \cdots, x_t = u_t}$. By this procedure, an asymmetric t-conference key distribution is exactly realized.

In this scheme, the required memory size for U_i is estimated as follows:

Theorem 1: The above scheme (Eq. (15)) is optimal in terms of memory size. *Namely, it achieves the lower* bound on $|U_i|$ (Lemma 2);

$$\log_2 |U_i| = \left(\prod_{j \in \{1,\cdots,t\}, j \neq i} (\psi_j + 1) \right) H(K). \tag{16}$$

For $t = 2$, we then obtain an optimal key sharing scheme for $< i_1, i_2 >= 1, < i_1, i_1 >= 0, < i_2, i_2 >= 0$ $(i_1 \neq i_2)$. When we apply this, the required memory for users is estimated as follows:

Corollary 1: The required memory for $u_{i_1} \in \mathcal{U}_{i_1}$ is

$$\log_2 |U_{i_1}| = |\mathcal{U}_{i_2}| H(K), \tag{17}$$

which is much less than that of normal KPSs as shown in Sect. 3.2.

5. Optimal Construction by Normal KPSs and Asymmetric 2-Conference Key Distribution Schemes

As already mentioned in Sect. 4.1, if optimal KPSs and optimal asymmetric 2-conference key distribution schemes are applied, the required memory size can be optimal. In this section, we show an optimal construction of a large-scale network by removing unnecessary communication links.

(1) **Procedure of the center:**

The center provides center algorithms of normal KPSs and asymmetric t-conference key distribution systems. For each $< i, i >= 1$ $(i = 1, 2, \cdots, N_{\mathcal{U}})$, a normal KPS is applied. And for each $< i, j >= 1$ $(i \neq j)$, an optimal asymmetric 2-conference key distribution scheme is applied. $G^i(x, y)$ and $G^{ij}(x, y)$ $(i, j \in \{1, 2, \cdots, N_{\mathcal{U}}\},\ i \neq j)$ denote the center algorithms for normal KPSs and asymmetric t-conference key distribution systems, respectively ($G^i(x, y)$ and $G^{ij}(x, y)$ holds $G^i(x, y) = G^i(y, x)$ and $G^{ij}(x, y) = G^{ji}(y, x)$, respectively). Then, the center gives the following secret algorithm U_{u_i} to $u_i \in \mathcal{U}_i$:

$$\begin{aligned} U_{u_i} = \{ & U^{ij}_{u_i}(y) \ | \text{For} < i, j >= 1, \\ & U^{ij}_{u_i}(y) = G^i(u_i, y) \ (i = j), \\ & U^{ij}_{u_i}(y) = G^{ij}(u_i, y) \ (i \neq j) \} \end{aligned} \tag{18}$$

Since applied KPSs (for $< i, i >= 1$) and asymmetric 2-conference key distribution schemes (for $< i, j >= 1, i \neq j$) are optimal, required memory size for each small key sharing system is estimated as follows:

$$\log_2 |U^{ij}_{u_i}(y)| = \begin{cases} (|\mathcal{U}_j| - 1) H(K) & (i = j) \\ |\mathcal{U}_j| H(K) & (i \neq j). \end{cases} \tag{19}$$

As an optimal KPS, Blundo et al.'s scheme, Matsumoto-Imai scheme and Blom's scheme are available. On the other hand, as an optimal asymmetric 2-conference key distribution scheme, our scheme presented in Sect. 4.3 is available.

(2) **Procedure of the entities:**

u_i computes the common key with $u_j \in \mathcal{U}_j$ as follows:

$$\begin{aligned} u_i &: k_{u_i, u_j} = U^{ij}_{u_i}(u_j), \\ u_j &: k_{u_i, u_j} = U^{ji}_{u_j}(u_i) (= U^{ij}_{u_i}(u_j)). \end{aligned} \tag{20}$$

6. Memory Size for Entities

In this section, we estimate the required memory size for users. Theorem 2 shows the amount of memory required for users' secret algorithms.

Theorem 2: By our construction, the required memory size for $u_i \in \mathcal{U}_i$ is estimated as follows:

$$\begin{aligned} & \log_2 |U_{u_i}| \\ & \quad = \left(\sum_{j \in \{1, \cdots, N_{\mathcal{U}}\}} (< i, j > |\mathcal{U}_j|) - < i, i > \right) \\ & \quad \cdot H(K). \end{aligned} \tag{21}$$

In addition, our scheme is optimal in terms of memory size.

Proof. Equation (21) is obviously obtained from Eq. (19). Additionally, this memory size is optimal due to Lemma 1. □

Here let $\log_2 |U'|$ be the required memory size for an entity when the whole network is constructed only by one normal KPS. Then, from Theorem 2 we obtain the following equation:

$$\begin{aligned} & \log_2 |U_{u_i}| \\ & = \frac{\sum_{j \in \{1, \cdots, N_{\mathcal{U}}\}} (< i, j > |\mathcal{U}_j|) - < i, i >}{\left(\sum_{j \in \{1, \cdots, N_{\mathcal{U}}\}} |\mathcal{U}_j| \right) - 1} \log_2 |U'|. \end{aligned} \tag{22}$$

Since $\sum_{j \in \{1, \cdots, N_{\mathcal{U}}\}} (< i, j > |\mathcal{U}_j|) - < i, i >$ is equal to the number of communication partners of u_i, and $\sum_{j \in \{1, \cdots, N_{\mathcal{U}}\}} |\mathcal{U}_j|$ is equal to the number of entities in the whole system, from Eq. (22) we have:

Theorem 3: Comparing with the required memory size for the conventional KPS, the required memory size for our scheme is estimated as follows:

$$\begin{aligned} & \log_2 |U_{u_i}| \\ & \quad = \frac{\text{the number of partners}}{\text{the number of entities} - 1} \log_2 |U'|. \end{aligned} \tag{23}$$

Namely, by using our scheme the memory size for an entity is reduced to be almost same as (# of partners)/(# of entities).

7. Modification for Not-Perfectly Secure Models

Our proposed scheme can be said perfectly secure since any subset of entities have no information on a key they should not know. When we reduce the collusion threshold for reduction of memory size, the security becomes inperfect. Namely, if the collusion threshold is less than the given one, by carrying out a collusion attack colluders can compute common keys of a victim. However, to succeed the attack, huge number of colluders will be required, and it seems still impossible to succeed collusion attacks in the real world if the collusion threshold is sufficiently large.

Generally, as described above, in such non-perfectly secure cases the system can be broken by collusion attacks if the number of colluders exceeds the collusion threshold. Namely, if the collusion threshold in entities $\mathcal{E}$, say $\omega_{\mathcal{E}}$, is less than $|\mathcal{E}|-1$, $\omega_{\mathcal{E}}+1$ colluders from $\mathcal{E}$ will be able to break the system. In this section, we show a modification of our proposed scheme to deal with the collusion attack more efficiently, assuming that the number of colluders is determined according to *Poisson distribution.*

Let p be the probability that a user joins a collusion attack. Assuming that p is constant among all users, the number of colluders is determined according to Poisson distribution. Hence, by a collusion attack in $\mathcal{E}$ the probability of attack success $P(\mathcal{E},\omega_{\mathcal{E}})$ is estimated as follows:

$$P(\mathcal{E},\omega_{\mathcal{E}}) = \sum_{i=\omega_{\mathcal{E}}+1}^{|\mathcal{E}|} e^{-p|\mathcal{E}|}\frac{(p|\mathcal{E}|)^i}{i!}. \qquad (24)$$

In this situation, we have

Lemma 3: We choose $\omega_{\mathcal{U}_i}$ $(i=1,\cdots,N_{\mathcal{U}})$ such that $P(\mathcal{U}_i,\omega_{\mathcal{U}_i}) = P$ $(0<P<1)$ and $\omega_{\mathcal{U}_i} > p|\mathcal{U}_i|$. Then, the following inequality holds

$$P(\mathcal{V},\omega_{\mathcal{V}}) \le P \quad \text{for all } \mathcal{V}, \qquad (25)$$

such that $\mathcal{V} = \{\mathcal{U}_{i_1},\mathcal{U}_{i_2},\cdots,\mathcal{U}_{i_{N_{\mathcal{V}}}}\}$, where $\{i_1,i_2,\cdots,i_{N_{\mathcal{V}}}\} \subseteq \{1,2,\cdots,N_{\mathcal{U}}\}$ and $\omega_{\mathcal{V}} = \left(\sum_{i\in\{i_1,\cdots,i_{N_{\mathcal{V}}}\}}\omega_i\right) + N_{\mathcal{V}} - 1$.

Proof. Regarding *the law of large number*, we obviously obtain the following inequality:

$$\begin{aligned}&P(\mathcal{X}\cap\mathcal{Y},\omega_{\mathcal{X}}+\omega_{\mathcal{Y}}+1)\\ &\quad \le \max\{P(\mathcal{X},\omega_{\mathcal{X}}),P(\mathcal{Y},\omega_{\mathcal{Y}})\} \text{ for all } \mathcal{X},\mathcal{Y}, \qquad (26)\end{aligned}$$

where $p|\mathcal{X}| < \omega_{\mathcal{X}} < |\mathcal{X}|-1$, $p|\mathcal{Y}| < \omega_{\mathcal{Y}} < |\mathcal{Y}|-1$. From Eq. (26), for the parameter setting such that for all i $P(\mathcal{U}_i,\omega_{\mathcal{U}_i}) = P$ $(0<P<1)$ and $\omega_{\mathcal{U}_i} > p|\mathcal{U}_i|$, Eq. (25) is obtained. □

Lemma 3 implies that the collusion threshold can be determined at a relatively small value if the number of entities is large. Namely, for the same security level the collusion threshold/the number of entities decreases as the number of entities increases. This fact can be formalized as follows:

Lemma 4: We choose $\omega_{\mathcal{V}}{}'$ such that $P(\mathcal{V}\omega_{\mathcal{V}}{}') = P$. Then, the following inequality holds

$$\omega_{\mathcal{V}}{}' \le \omega_{\mathcal{V}} \quad \text{for all } \mathcal{V}. \qquad (27)$$

Proof. Since $P(\mathcal{V},\omega_{\mathcal{V}}) \le P$ (Lemma 3), the collusion threshold in $\mathcal{V}$ which allows the probability of attack success $= P$ is less than $\omega_{\mathcal{V}}$. □

In the proposed key-sharing scheme, for communication between $u_i \in \mathcal{U}_i$ and $u_j \in \mathcal{U}_j, j \in \mathcal{J}'$ for all $\mathcal{J}'$, such that $\mathcal{J}' \subseteq \mathcal{J} = \{j| < i,j >= 1, j \ne i\}$, applied asymmetric 2-conference key distribution systems are established separately. By combining them into one system, we can reduce the memory size for u_i. As described in Eq. (18), U_{u_i} consists of $U_{u_i}^{ij}(y)$ for all j such that $< i,j >= 1$. Recall that $U_{u_i}^{ij}(y)$ is u_i's secret algorithm of the asymmetric 2-conference key distribution scheme for communication between $\mathcal{U}_i$ and $\mathcal{U}_j$. Since asymmetric 2-conference key distribution systems for communication between $\mathcal{U}_i$ and $\mathcal{U}_j, j \in \mathcal{J}' \subseteq \mathcal{J} = \{j| < i,j >= 1, j \ne i\}$ can be replaced with an asymmetric 2-conference key distribution system for communication between $\mathcal{U}_i$ and $\{\mathcal{U}_j | j \in \mathcal{J}'\}$. Here, we call this replacement *combining.*

Theorem 4: If a system allows combining $U_{u_i}^{ij}(y), j \in \mathcal{J}'$ into $U_{u_i}^{i\mathcal{J}'}(y)$ which is an optimal asymmetric 2-confernce key distribution system between $\mathcal{U}_i$ and all $\mathcal{U}_j, j \in \mathcal{J}'$, for the same security level (the probability of attack success is P) the required memory size is reduced as follows:

$$\log_2|U_{u_i}^{i\mathcal{J}'}(y)| = \frac{\omega_{\mathcal{J}'}{}'+1}{\omega_{\mathcal{J}'}+1}\sum_{j\in\mathcal{J}'}\log_2|U_{u_i}^{ij}(y)|, \qquad (28)$$

where for all $j \in J'$ $P(\mathcal{U}_j,\omega_{\mathcal{U}_j}) = P$, $\omega_{\mathcal{U}_j} > p|\mathcal{U}_j|$, $\omega_{\mathcal{J}'} = \left(\sum_{j\in\mathcal{J}'}\omega_{\mathcal{U}_j}\right) + |\mathcal{J}'| - 1$ and $P(\mathcal{J}',\omega_{\mathcal{J}'}{}') = P$.

Proof. Since $\log_2|U_{u_i}^{i\mathcal{J}'}(y)| = (\omega_{\mathcal{J}'}{}'+1)H(K)$ and $\sum_{j\in\mathcal{J}'}\log_2|U_{u_i}^{ij}(y)| = \left(\sum_{j\in\mathcal{J}'}(\omega_{\mathcal{U}_j}+1)\right)H(K)$, Eq. (28) is obtained. Note that $\sum_{j\in\mathcal{J}'}(\omega_{\mathcal{U}_j}+1) = \omega_{\mathcal{J}'}+1$. In addition, the memory size is exactly reduced due to Lemma 4. □

8. Conclusion

In this paper, an optimal construction of ID-based key sharing scheme for large-scale networks is proposed. It has been pointed out that in order to achieve perfect security a huge amount of memory is required in conventional KPS, and it has been shown how KPS can be improved for practical communication systems. By removing communication links that are not required in a practical communication system. This approach yields that the amount of memory is reduced significantly. In our scheme, the required memory for each entity is (the number of partners) × (the length of a common key), while in the conventional KPS is (the number of entities −1) × (the length of a common key).

So, if an entity communicates only with $1/r$ of others, the required memory is reduced almost $1/r$ of that required in conventional KPS. Furthermore, our scheme is proven to be optimal. The previous makes our scheme attractive for various applications like broadcasting or E-commerce in the Internet. In this paper, we also propose an optimal asymmetric t-conference key distribution scheme. Since this scheme has good properties, it could be utilized effectively in other applications. Additionally, since public-key cryptosystems have no advantages over KPS in terms of computational costs, ID-basedness, and so on, the efficient combination of a public-key cryptosystem and our scheme will yield a more efficient and secure communication system in comparison with single use of a public-key cryptosystem.

References

[1] T. Matsumoto and H. Imai, "On the key predistribution system: A practical solution to the key distribution problem," Proc. CRYPTO'87, LNCS 293, pp.185–193, Springer-Verlag, 1987.

[2] R. Blom, "Non-public key distribution," Proc. CRYPTO'82, pp.231–236, Plenum Press, 1983.

[3] A. Fiat and A. Shamir, "How to prove yourself: Practical solutions to identification and signature problems," Proc. CRYPTO'86, LNCS 263, pp.186–194, Springer-Verlag, 1986.

[4] A. Shamir, "Identity-based cryptosystems and signature schemes," Proc. CRYPTO'84, LNCS 196, pp.47–53, Springer-Verlag, 1985.

[5] U. Maurer and Y. Yacobi, "Non-interactive public-key cryptography," Proc. Eurocrypt'91, LNCS 547, pp.498–407, Springer-Verlag, 1992.

[6] U. Maurer and Y. Yacobi, "A remark on a non-interactive public-key distribution system," Proc. Eurocrypt'92, LNCS 658, pp.458–460, Springer-Verlag, 1993.

[7] E. Okamoto and K. Tanaka, "Identity-based information security management system for personal comuputer networks," IEEE J. Sel. Areas Commun., vol.7, no.2, pp.290–294, 1989.

[8] H. Tanaka, "A realization scheme of the identity-based cryptosystems," Proc. CRYPTO'87, LNCS 293, pp.340–349, Springer-Verlag, 1988.

[9] S. Tsujii and J. Chao, "A new ID-based key sharing system," Proc. CRYPTO'91, LNCS 576, pp.288–299, Springer-Verlag, 1992.

[10] D. Coppersmith, "Attack on the cryptographic scheme NIKS-TAS," Proc. CRYPTO'94, LNCS 839, pp.40–49, Springer-Varlag, 1994.

[11] L. Gong and D.J. Wheeler, "A matrix key-distribution scheme," J. Cryptology, vol.2, pp.51–59, Springer-Verlag, 1993.

[12] W.A. Jackson, K.M. Martin, and C.M. O'Keefe, "Multi-secret threshold schemes," Proc. CRYPTO'93, LNCS 773, pp.126–135, Springer-Verlag, 1994.

[13] Y. Desmedt and V. Viswanathan, "Unconditionally secure dynamic conference key distribution," IEEE, ISIT'98, 1998.

[14] C. Blundo, A. De Santis, A. Herzberg, S. Kutten, U. Vaccaro, and M. Yung, "Perfectly secure key distribution for dynamic conferences," Proc. CRYPTO'92, LNCS 740, pp.471–486, Springer-Verlag, 1993.

[15] C. Blundo, L.A. Frota Mattos, and D.R. Stinson, "Trade-offs between communication and strage in unconditionally secure schemes for broadcast encryption and interactive key distribution," Proc. CRYPTO'96, LNCS 1109, pp.387–400, Springer-Verlag, 1996.

[16] A. Fiat and M. Naor, "Broadcast encryption," Proc. CRYPTO'93, LNCS 773, pp.480–491, Springer-Verlag, 1994.

[17] G. Hanaoka, T. Nishioka, Y. Zheng, and H. Imai, "An efficient hierarchical identity-based key-sharing method resistant against collusion-attacks," Proc. Asiacrypt'99, LNCS 1716, pp.348–362, Springer-Verlag, 1999.

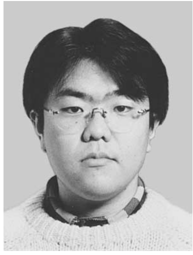
[18] K. Kurosawa, K. Okada, H. Saido, and D. Stinson, "New combimatorial bounds for authentication codes and key predistribution schemes," Designs, Codes and Cryptography, vol.15, pp.87–100, 1998.

[19] K. Kurosawa, T. Yoshida, Y. Desmedt, and M. Burmester, "Some bounds and a construction for secure broadcast encryption," Proc. Asiacrypt'98, LNCS 1514, pp.420–433, Springer-Verlag, 1998.

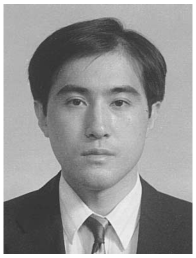
[20] G. Hanaoka, T. Nishioka, Y. Zheng, and H. Imai, "An optimization of credit-based payment for electronic toll collection systems," IEICE Trans., vol.E83-A, no.8, pp.1681–1690, 2000.

[21] T. Matsumoto, Y. Takashima, H. Imai, M. Sasaki, H. Yoshikawa, and S. Watanabe, "A prototype KPS and its application—IC card based key sharing and cryptographic communication," IEICE Trans., vol.E73, no.7, pp.1111–1119, July 1990.

[22] G. Hanaoka, T. Nishioka, Y. Zheng, and H. Imai, "Optimal construction of unconditionally secure ID-based key sharing scheme for large-scale networks," Proc. The Second International Conference on Information and Communication Security (ICICS'99), LNCS 1726, pp.157–168, Springer-Verlag, 1999.



Goichiro Hanaoka is currently a Ph.D. student in the Information and Communication Engineering Department at the University of Tokyo, Tokyo, Japan. He has received his bachelors and masters degrees in Electronic engineering and Information and communication engineering from the University of Tokyo in 1997 and 1999, respectively. He was awarded the excellent paper prize from SITA in 2000. His research interests are in the fields of cryptography, electronic payments and network security. He is a Research Fellow of Japan Society for the Promotion of Science (JSPS).



Tsuyoshi Nishioka was born in Tokyo, Japan on March 21, 1965. He recieved the B.S., M.S. and, Ph.D. degrees in physics from the University of Tokyo, Tokyo, in 1987, 1989 and 1992, respectively. He is currently a researcher of Information Technology R&D Center in Mitsubishi Electric Corporation. His current research interests are cryptography and its applications. He is a member of the Physical Society of Japan and a member of the Institute of Electronics, Information and Communication Engineers.

Yuliang Zheng received his B.Sc. degree in computer science from Nanjing Institute of Technology, China, in 1982, and the M.E. and Ph.D. degrees, both in electrical and computer engineering, from Yokohama National University, Japan, in 1988 and 1991 respectively. From 1982 to 1984 he was with the Guangzhou Research Institute for Communications, Guangzhou (Canton), China. Since 1991 he has worked for a number of academic institutions in Australia. Currently he is Reader of the Faculty of Information Technology, Monash University, in Melbourne, and heads Monash's Laboratory for Information and Network Security (LINKS). He served as the program committee co-chair of the 1998, 1999 and 2000 International Workshops on Practice and Theory in Public Key Cryptography. His research interests include cryptography and its applications secure electronic commerce. Dr. Zheng is a member of IACR, ACM and IEEE.

Hideki Imai was born in Shimane, Japan on May 31, 1943. He received the B.E., M.E., and Ph.D. degrees in electrical engineering from the University of Tokyo in 1966, 1968, 1971, respectively. From 1971 to 1992 he was on the faculty of Yokohama National University. In 1992 he joined the faculty of the University of Tokyo, where he is currently a Full Professor in the Institute of Industrial Science. His current research interests include information theory, coding theory, cryptography, spread spectrum systems and their applications. He received Excellent Book Awards from IEICE in 1976 and 1991. He also received the Best Paper Award (Yonezawa Memorial Award) from IEICE in 1992, the Distinguished Services Award form the Association for Telecommunication Promotion Month in 1994, the Telecom System Technology Prize from the Telecommunication Advancement Foundation and Achievement Award from IEICE in 1995. In 1998 he was awarded Golden Jubilee Paper Award by the IEEE Information Theory Society. In 1999 he was awarded Honor Doctor Degree from Soonchunhyang University, Korea. He was elected an IEEE Fellow for his contributions to the theory of coded modulation and two-dimensional codes in 1992. He chaired many committees of scientific societies such as the IEICE Professional Group on Information Theory and many international conferences such as ITW'99 (1993 IEEE Information Theory Workshop), ISITA'94 (1994 International Symposium on Information Theory and Its Applications), and ISITA'96. He also created several series of conferences such as SCIS (Symposium on Cryptography and Information Security: The largest series of conferences on information security in Japan), PKC (International Workshop on Practice and Theory in Public Key Cryptography) and WPMC (International Symposium on Wireless Personal Multimedia Communications). He served as the editor for several scientific journals of IEICE, IEEE etc. Dr. Imai has been on the board of IEICE, the IEEE Information Theory Society, Japan Society of Security Management (JSSM) and the Society of Information Theory and Its Applications (SITA). He served as President of the IEICE Engineering Sciences Society and SITA.