A Hierarchical Non-interactive Key-Sharing Scheme with Low Memory Size and High Resistance against Collusion Attacks

GOICHIRO HANAOKA[1]*, TSUYOSHI NISHIOKA[2], YULIANG ZHENG[3] AND HIDEKI IMAI[1]

[1]*Information and Systems, Institute of Industrial Science, the University of Tokyo, Komaba 4-6-1, Meguro-ku, Tokyo 153-8505, Japan*
[2]*Information Technology R&D Center, Mitsubishi Electric Corporation, Japan*
[3]*Department of Software and Information Systems, University of North Carolina at Charlotte, USA*
Email: hanaoka@imailab.iis.u-tokyo.ac.jp

Efficient ID-based key sharing schemes are desired worldwide for secure communications on Internet and other networks. The Key Predistribution Systems (KPSs) are a large class of such key sharing schemes. The remarkable property of KPSs is that in order to share the key, a participant should only input its partner's identifier to its secret KPS algorithm. Although it has many advantages in terms of efficiency, on the other hand it is vulnerable to certain collusion attacks. While conventional KPSs establish communication links between any pair of entities in a communication system, in many practical communication systems, such as broadcasting, not all links are required. In this paper, we propose a new version of KPS which is called the *Hierarchical KPS*. In the Hierarchical KPS, simply by removing unnecessary communication links, we can significantly increase the collusion threshold. As an example, for a typical security parameter setting, the collusion threshold of the Hierarchical KPS is 16 times higher than that of the conventional KPS while using the same amount of memory at the KPS center. The memory required by the user is even reduced by a factor 1/16 in comparison with the conventional linear scheme. Hence, Hierarchical KPS provides a more efficient method for secure communication.

Received 16 March 2000; revised 12 September 2001

1. INTRODUCTION

For information security, ID-based key distribution technologies are quite important. The concept of ID-based key cryptosystems was originally proposed by Fiat and Shamir [1, 2]. Maurer and Yacobi then presented an ID-based key distribution scheme following Shamir's concept [3, 4]. However, their scheme required a huge computational power. Okamoto and Tanaka [5] also proposed a key distribution scheme based on a user's identifier, but it required prior communications between a sender and a receiver to share the employed key. Although Tsujii and others proposed several ID-based key distribution schemes [6, 7], almost all of them had been broken [8]. Thus, the performance of these schemes is unsatisfactory. However, Blom's ID-based key distribution scheme [9], which is generalized by Matsumoto and Imai [10], cannot be overlooked, especially in terms of computational complexity and non-interactivity. Many useful schemes based on Blom's scheme have been proposed [10, 11, 12, 13, 14, 15] which are known as *Key Predistribution Systems* (KPSs).

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 partner's identifier to its secret KPS algorithm. Blundo *et al.* [14, 16] showed a lower bound of memory size of users' KPS algorithms and developed a KPS for conference key distribution. Moreover, Fiat and Naor [15] and Kurosawa *et al.* [17] applied KPSs to a broadcasting encryption system.

Although a KPS has many desired properties, it too has the following problem: when a number of users, which exceeds a certain threshold, cooperate they can calculate the central authority's secret information. Setting up a higher collusion threshold in this scheme requires larger amounts of memory in the center as well as for the users. The solution of this problem will make KPSs much more attractive for ID-based key distribution.

Although KPSs provide common keys for all possible communication links among entities, in practical communication systems most of them will not be necessary.

By removing these unnecessary communication links, we can increase the collusion threshold significantly. This will be explained by means of a new version of the KPS called the *Hierarchical KPS*. A Hierarchical KPS demonstrates how to optimize a KPS for a communication system against collusion attacks. The Hierarchical KPS is constructed based on the Matsumoto–Imai scheme [10]. Since the key distribution procedure in the Matsumoto–Imai scheme consists of simple calculations only, computational cost in a Hierarchical KPS can also be set to be quite small. As an example, for a typical security parameter setting, the collusion threshold of a Hierarchical KPS is 16 times higher than that of a conventional KPS while using the same amount of memory in the KPS center. The memory required by the user can even be reduced to $1/16$ of that of a conventional KPS.

Section 2 gives a brief review of the KPS. In Section 3, the Hierarchical KPS is introduced. This is followed by the evaluation and discussion of the security of Hierarchical KPSs in Section 4. Section 5 closes the paper with some concluding remarks.

2. A BRIEF OVERVIEW OF KPSs

2.1. Key Predistribution System

A KPS consists of two kinds of entities: the KPS center and the users who want to share a common key. The KPS center possesses a secret algorithm by which it can generate an individual KPS 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.

The KPS is generalized as follows [10]. First the KPS center produces a random *symmetric* function $f(x, y)$, which is called the *KPS-center algorithm*. $f(x_A, y)$ is given to user A as his *secret KPS algorithm* by the KPS center, where x_A indicates the effective ID of A. When users A and B want to set up a cryptographic communication, they can share a common key $f(x_A, x_B)$ by inputting their communication partners' identifiers to their secret KPS algorithms.

2.2. Matsumoto–Imai scheme [10]

This subsection explains how the users' secret KPS algorithms are generated and how users share a common key in the manner of the Matsumoto–Imai scheme. Note that all the calculations in this paper are related to the finite field $GF(2)$.

Let the m-dimensional vectors x_A and x_B be the effective IDs of entities A and B, respectively. The $(m \times m)$ symmetric matrices $G^{(\mu)}$ $(\mu = 1, \ldots, h)$ are KPS-center algorithms. The $G^{(\mu)}$s are produced by the KPS center and kept secret from all other entities. $G^{(\mu)}$ generates the μth bit of a communication key between users A and B, so h is the length of this key. $X_A^{(\mu)}$ and $X_B^{(\mu)}$ are the secret KPS algorithms of A and B, respectively. $X_A^{(\mu)}$ and $X_B^{(\mu)}$ are calculated by the KPS center as follows:

$$X_A^{(\mu)} = x_A\, G^{(\mu)}, \quad X_B^{(\mu)} = x_B\, G^{(\mu)}.$$

$X_A^{(\mu)}$ and $X_B^{(\mu)}$ are contained in *tamper-resistant modules* (TRMs) and distributed to A and B, respectively. (If procedures for inputting data into a TRM are thought to be complicated, a TRM is not necessary.) By using $X_A^{(\mu)}$ and $X_B^{(\mu)}$, A and B share their symmetric key as follows:

$$A:\ k_{AB}^{(\mu)} = X_A^{(\mu)}\,{}^{\mathrm{t}}x_B, \quad B:\ k_{AB}^{(\mu)} = X_B^{(\mu)}\,{}^{\mathrm{t}}x_A,$$

where $k_{AB}^{(\mu)}$ indicates the μth bit of the shared key k_{AB} between A and B.

2.3. Property and problem of KPSs

KPSs, including the Matsumoto–Imai scheme, have 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, the key distribution procedure consists of simple calculations so that computational costs are quite small. Finally, in order to share the key, a participant has only to input its partner's identifier to its secret KPS algorithm. Thus, KPSs are very applicable to one-pass or quick-response transactions, e.g. mail systems, broadcasting systems, electronic toll collection systems and so on.

However, a KPS has a certain collusion threshold; when more users cooperate they can calculate the KPS-center algorithm $G^{(\mu)}$. For example in the Matsumoto–Imai scheme, as already mentioned above $G^{(\mu)}$ is an $(m \times m)$ matrix. Hence, by using m linearly independent secret KPS algorithms, the KPS-center algorithm can be easily revealed (note however that, in order to participate in this collusion attack, each adversary has to break his TRM). Thus, m is determined by the number of users. In order to avoid such collusion attacks, we need to increase the value of m. However, since the number of elements of $G^{(\mu)}$ is m^2, a quite large memory size is required for the KPS center in order to increase the value of m. Furthermore, the memory size of a user's secret KPS algorithm is thereby enlarged in proportion to m. Although these memory sizes are not small, they are proven to be optimal [14]. Therefore, in the conventional KPS, we cannot cope with collusion attacks efficiently. This can be a serious problem, especially in a situation where the available memory is strictly limited (e.g. IC cards). For example, $m = 8192$ is selected as the collusion threshold in the 'KPSL1 card' [18], where the key length is 64 bits. The secret algorithm itself then consumes 64 kbytes of memory size for each IC card. Therefore, a KPS was considered to be somewhat expensive for realistic IC card systems at that time. Furthermore, by introducing 128–256 bit symmetric key cryptosystems, the required memory size will be 128–256 kbytes.

Although the conventional KPS provides a common key between any pair of entities, most of them are not necessary in practical communication systems. When no keys are

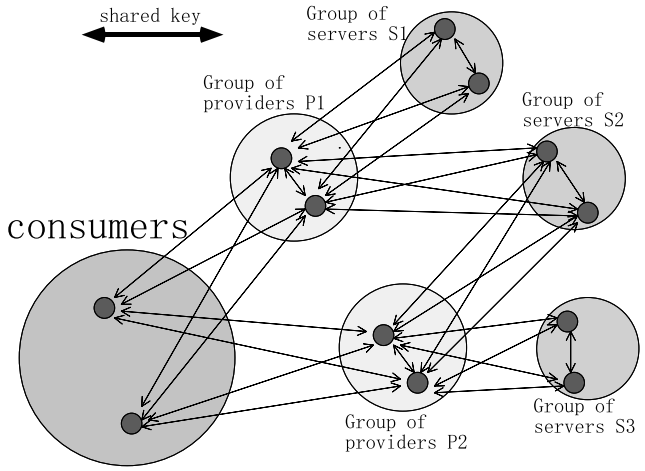

FIGURE 1. Communication links in a Hierarchical KPS.

provided for such unnecessary communication links, the collusion threshold can be increased and the memory size of the users decreased, while the memory size of the KPS center stays the same.

3. HIERARCHICAL KPS

In practical communication systems, such as broadcasting, entities are classified into three classes: *consumer*, *provider* and *server*. Figure 1 displays the structure of their communication links. Consumers, i.e. the majority of entities, receive information from any provider. Servers hold information required by providers. For example, in broadcasting, addressees and broadcasting stations are regarded as consumers and providers, respectively. Certain entities that provide information for broadcasting stations are regarded as servers. In such a communication structure, communication links between consumers are not necessary. Only communication links to providers are required for the consumers. Similarly, although some communication links between providers and servers are required, not all of them are necessary. Furthermore, although communication links among providers/servers are required, not all of them are necessary. So, providers and servers can be divided into multiple groups. Then, we should realize the possibility to share a common key only for:

- links between consumers and providers;
- links among providers who belong to the same group;
- links among servers who belong to the same group;
- links between providers and servers, assuming the group of providers and the group of servers are allowed to communicate with each other.

Necessary communication links in this structure are summarized in Table 1.

As mentioned above, in the Matsumoto–Imai scheme the collusion threshold can be increased by replacing the square $(m \times m)$ matrix $G^{(\mu)}$ of the center algorithm by a larger square matrix; this however requires a significantly larger memory size in the KPS center. Another possibility is to

TABLE 1. Required communications in practical communication systems, where ◯, △ and × indicate required, partly required and unnecessary, respectively.

	Consumer	Provider	Server
Consumer	×	◯	×
Provider	◯	△	△
Server	×	△	△

replace the $(m \times m)$ square matrix by a rectangular $(m' \times n')$ matrix of the same size, $m^2 \simeq m' \times n'$, $m' > m$, $n' < m$. This requires the set of users to be split into two distinct subsets. The threshold for a collusion of members of the first subset is m' and that of the second subset is n'. Then a member of one subset can share a common key only with any of the members of the other subset; common keys between members of the same subset are not possible. This type of KPS with asymmetric center algorithm will be used below to realize key distribution between consumers and providers, since no common keys are required among consumers.

From the requirement that the memory size of the KPS center should be fixed, i.e. from the equation $m^2 \simeq m' \times n'$, it becomes clear that the collusion threshold m' for the consumers will increase when n' decreases. This means that there should only be a few members in the second subset. Therefore, a member of this subset is a group of providers who all provide access to several groups of servers. In other words, a 'layer' of provider groups is inserted between the consumers and the groups of servers, see Figure 1. Therefore, the new version of a KPS is called a 'Hierarchical KPS'. The following sections explain it in detail.

3.1. Key distribution between consumers and providers

The improvement that we have made in a KPS starts with replacing the symmetric matrices for the KPS-center algorithm in the Matsumoto–Imai scheme with asymmetric $(m \times n)$ matrices $G^{(\mu)}$ $(\mu = 1, \ldots, h,\ m \geq n)$. Then key distribution between consumers and providers is implemented in the following way.

Let the m-dimensional vector x_C be the effective ID of consumer C and the n-dimensional vector y_P be the effective ID of provider P. Then C's secret KPS algorithm $X_C^{(\mu)}$ is calculated by

$$X_C^{(\mu)} = x_C\, G^{(\mu)},$$

and $Y_P^{(\mu)}$ is P's secret KPS algorithm which is calculated as follows:

$$Y_P^{(\mu)} = y_P\, {}^{\mathrm{t}}G^{(\mu)}.$$

C and P share their symmetric key k_{PC} according to

$$C:\ k_{PC}^{(\mu)} = x_C\, G^{(\mu)}\, {}^{\mathrm{t}}y_P = X_C^{(\mu)}\, {}^{\mathrm{t}}y_P,$$
$$P:\ k_{PC}^{(\mu)} = y_P\, {}^{\mathrm{t}}G^{(\mu)}\, {}^{\mathrm{t}}x_C = Y_P^{(\mu)}\, {}^{\mathrm{t}}x_C,$$

$$k_{CP}^{(\mu)} = \overset{x_C}{\begin{pmatrix} 1 & 1 & 0 & \cdots & 0 & 1 \end{pmatrix}} \overset{G^{(\mu)}}{\begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \\ 0 & 1 & \cdots & 0 \end{pmatrix}} \overset{{}^t y_P}{\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{pmatrix}} = \overset{X_C^{(\mu)}}{\begin{pmatrix} 1 & 1 & \cdots & 0 \end{pmatrix}} \overset{{}^t y_P}{\begin{pmatrix} 1 \\ 0 \\ \vdots \\ 1 \end{pmatrix}}$$

$$k_{CP}^{(\mu)} = \overset{y_P}{\begin{pmatrix} 1 & 0 & \cdots & 1 \end{pmatrix}} \overset{{}^t G^{(\mu)}}{\begin{pmatrix} 1 & 0 & 1 & \cdots & 1 & 0 \\ 0 & 0 & 1 & \cdots & 1 & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 1 & 1 & \cdots & 1 & 0 \end{pmatrix}} \overset{{}^t x_C}{\begin{pmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}} = \overset{Y_P^{(\mu)}}{\begin{pmatrix} 0 & 1 & 0 & \cdots & 1 & 0 \end{pmatrix}} \overset{{}^t x_C}{\begin{pmatrix} 1 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}}$$

FIGURE 2. Key distribution between a consumer and a provider in a Hierarchical KPS.

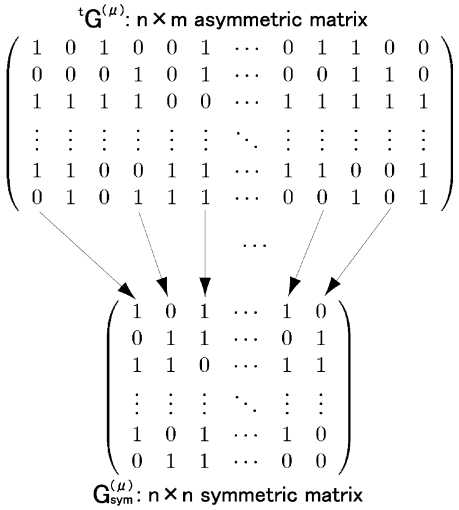

FIGURE 3. The embedded symmetric matrix $G_{\mathrm{sym}}^{(\mu)}$ in $G^{(\mu)}$.

where $k_{CP}^{(\mu)}$ indicates the μth bit of k_{CP}, the shared key between C and P. Figure 2 illustrates the key distribution between a consumer and a provider in a Hierarchical KPS.

3.2. Key distribution among providers

In this subsection, we explain key distribution among providers with asymmetric matrices $G^{(\mu)}$.

For key distribution among providers we embed a symmetric matrix $G_{\mathrm{sym}}^{(\mu)}$ in $G^{(\mu)}$, where $G_{\mathrm{sym}}^{(\mu)}$ consists of rows in $G^{(\mu)}$ as shown in Figure 3. Therefore, it is obvious that $G_{\mathrm{sym}}^{(\mu)}$ can be embedded in $G^{(\mu)}$ if $m \geq n$. By using $G_{\mathrm{sym}}^{(\mu)}$, providers can share their keys as shown in Figure 4. According to the selection of rows belonging to $G_{\mathrm{sym}}^{(\mu)}$ in $G^{(\mu)}$, elements from $Y_P^{(\mu)}$ and $Y_{P'}^{(\mu)}$ are selected to form n-dimensional vectors $\overline{Y_P^{(\mu)}}$ and $\overline{Y_{P'}^{(\mu)}}$. Using $\overline{Y_P^{(\mu)}}$ and $\overline{Y_{P'}^{(\mu)}}$, two providers P and P' share their key as follows:

$$P:\ k_{PP'}^{(\mu)} = \overline{Y_P^{(\mu)}}\,{}^t y_{P'}, \quad P':\ k_{PP'}^{(\mu)} = \overline{Y_{P'}^{(\mu)}}\,{}^t y_P.$$

Again, $k_{PP'}^{(\mu)}$ indicates the μth bit of the shared key $k_{PP'}$ between P and P'. We should note here that the usage of the symmetric matrix is almost the same as that for key distribution in a conventional KPS.

Although providers can share their keys by using this method, it also has the following problems:

- $G_{\mathrm{sym}}^{(\mu)}$ might be revealed by consumers' collusion attacks, if the selection of rows belonging to $G_{\mathrm{sym}}^{(\mu)}$ in $G^{(\mu)}$ has been exposed (for convenience, call this selection $k_{\mathrm{sel}}^{(\mu)}$);
- the length of a key between two providers may not be longer than a key between a provider and a consumer.

As already mentioned, we assume that there are some groups of providers where each provider communicates only with other providers in his group. The above problem can be solved if more than one $G_{\mathrm{sym}}^{(\mu)}$ is extracted from one $G^{(\mu)}$ and more than one $k_{\mathrm{sel}}^{(\mu)}$ is distributed to each group of providers.

Suppose that $G_{\mathrm{sym}}^{(\mu),ij}$ $(i = 1, \ldots, N_{\mathrm{sym}},\ j = 1, \ldots, N_P)$ are $n \times n$ symmetric matrices embedded in $G^{(\mu)}$ and $k_{\mathrm{sel}}^{(\mu),ij}$ $(i = 1, \ldots, N_{\mathrm{sym}},\ j = 1, \ldots, N_P)$ are a selection of rows belonging to $G_{\mathrm{sym}}^{(\mu),ij}$ in $G^{(\mu)}$. N_{sym} is the number of embedded symmetric matrices that are distributed to one group of providers and N_P is the number of groups of providers. Here, each of $G_{\mathrm{sym}}^{(\mu),ij}$ $(i = 1, \ldots, N_{\mathrm{sym}},\ j = 1, \ldots, N_P)$ is embedded in $G^{(\mu)}$ as shown in Figure 3 and each of the rows in $G^{(\mu)}$ belongs to one of $G_{\mathrm{sym}}^{(\mu),ij}$ $(i = 1, \ldots, N_{\mathrm{sym}},\ j = 1, \ldots, N_P)$ at the most. Note that $N_{\mathrm{sym}}N_P$ should not be more than m/n considering the security of the system. $k_{\mathrm{sel}}^{(\mu),ij}$ $(i = 1, \ldots, N_{\mathrm{sym}})$ are distributed to all providers in the jth group $\mathcal{P}_j$. Then, key distribution between providers P and P', where both belong to $\mathcal{P}_j$, is carried out as follows:

$$P:\ k_{PP'}^{(\mu),ij} = \overline{Y_P^{(\mu),ij}}\,{}^t y_{P'} \quad (i = 1, \ldots, N_{\mathrm{sym}}),$$
$$P':\ k_{PP'}^{(\mu),ij} = \overline{Y_{P'}^{(\mu),ij}}\,{}^t y_P \quad (i = 1, \ldots, N_{\mathrm{sym}}),$$

where $k_{PP'}^{(\mu),ij}$ $(i = 1, \ldots, N_{\mathrm{sym}})$ indicates the μth bit of

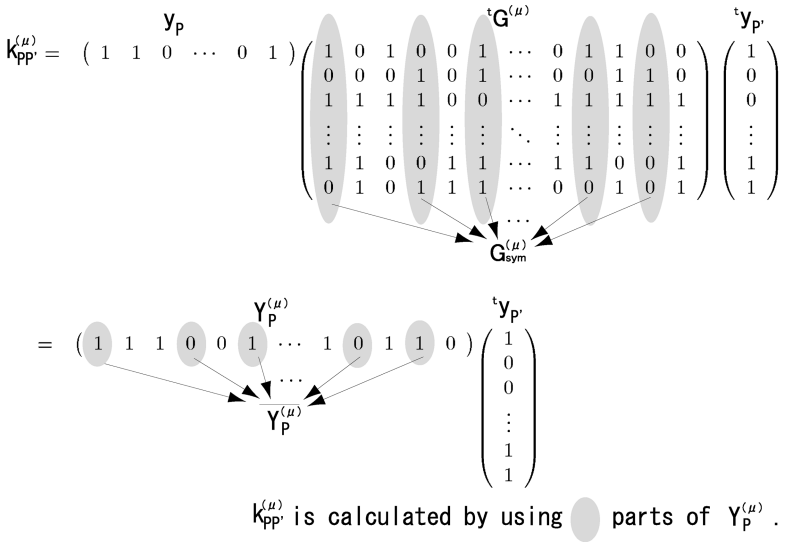

FIGURE 4. Key distribution among providers.

the shared key $k^{ij}_{PP'}$ $(i = 1, \ldots, N_{\mathrm{sym}})$ between P and P', and elements from Y_P and $Y_{P'}$ are selected according to $k^{(\mu),ij}_{\mathrm{sel}}$ $(i = 1, \ldots, N_{\mathrm{sym}})$ to form n-dimensional vectors $\overline{Y^{(\mu),ij}_P}$ and $\overline{Y^{(\mu),ij}_{P'}}$ $(i = 1, \ldots, N_{\mathrm{sym}})$.

So, if a $G^{(\mu),i_0 j}_{\mathrm{sym}}$ has been exposed by a certain consumer's attack, the providers in $\mathcal{P}_j$ can deal with this attack by using another $k^{(\mu),i_1 j}_{\mathrm{sel}}$, $i_1 \neq i_0$. Furthermore, by using multiple $k^{(\mu)}_{\mathrm{sel}}$ simultaneously, providers can share longer keys. For example, if both $k^{(\mu),i_0 j}_{\mathrm{sel}}$ and $k^{(\mu),i_1 j}_{\mathrm{sel}}$ are used simultaneously, the length of the keys among providers in $\mathcal{P}_j$ can be $2h$, which is twice the length of the keys between consumers and providers. Accordingly, the keys shared among providers can be at most $N_{\mathrm{sym}}h$.

Additionally, note that this scheme permits a provider to belong to multiple groups concurrently.

3.3. Key distribution between providers and servers

As already mentioned, servers can share keys with providers, assuming that the groups they belong to are allowed to communicate with each other. In this subsection, we show how to produce a server's secret KPS algorithm.

Let the n-dimensional vectors z_S be the effective ID of server S and let $Z^{(\mu),ij}_S$ be the secret KPS algorithm of S which is calculated as follows:

$$Z^{(\mu),ij}_S = z_S\, G^{(\mu),ij}_{\mathrm{sym}} \quad (i = 1, \ldots, N_{\mathrm{sym}}).$$

Herein it is assumed that S belongs to a group of servers $\mathcal{S}_j$ that is allowed to communicate with the providers in group $\mathcal{P}_j$. By using this secret KPS algorithm, communication keys are shared between S and P as follows:

$$S:\ k^{(\mu),ij}_{SP} = Z^{(\mu),ij}_S\, {}^t y_P \quad (i = 1, \ldots, N_{\mathrm{sym}}),$$
$$P:\ k^{(\mu),ij}_{SP} = \overline{Y^{(\mu),ij}_P}\, {}^t z_S \quad (i = 1, \ldots, N_{\mathrm{sym}}),$$

where $k^{(\mu),ij}_{SP}$ $(i = 1, \ldots, N_{\mathrm{sym}})$ indicates the μth bit of the shared key k^{ij}_{SP} $(i = 1, \ldots, N_{\mathrm{sym}})$ between S and P.

Similarly to the key distribution among providers, even if $G^{(\mu),i_0 j}_{\mathrm{sym}}$ is exposed by a certain attack, S and P can still share their key using other $Z^{(\mu),i_1 j}_S$ and $\overline{Y^{(\mu),i_1 j}_P}$, $i_1 \neq i_0$. Moreover, by concurrent use of their secret KPS algorithms again, longer keys can be used. For example, if $Z^{(\mu),i_0 j}_S$, $Z^{(\mu),i_1 j}_S$ and $\overline{Y^{(\mu),i_0 j}_P}$, $\overline{Y^{(\mu),i_1 j}_P}$ are used, the length of a shared key will be $2h$. As mentioned above, the maximum length of the key shared between providers and servers in this manner can be $N_{\mathrm{sym}}h$.

Note that a group of servers can be allowed to communicate with multiple groups of providers in this way and that a server can belong to multiple groups of servers.

3.4. Key distribution among servers

Any pair of servers in the same group can share their communication key using the servers' secret KPS algorithms mentioned in Section 3.3. Namely, a pair of servers S and S', belonging to $\mathcal{S}_j$, share their common key as follows:

$$S:\ k^{(\mu),ij}_{SS'} = Z^{(\mu),ij}_S\, {}^t z_{S'} \quad (i = 1, \ldots, N_{\mathrm{sym}}),$$
$$S':\ k^{(\mu),ij}_{SS'} = Z^{(\mu),ij}_{S'}\, {}^t z_S \quad (i = 1, \ldots, N_{\mathrm{sym}}),$$

where $k^{(\mu),ij}_{SS'}$ $(i = 1, \ldots, N_{\mathrm{sym}})$ indicates the μth bit of the shared key $k^{ij}_{SS'}$ $(i = 1, \ldots, N_{\mathrm{sym}})$ between S and S', $z_{S'}$ is the effective ID of S' and $Z^{(\mu),ij}_{S'}$ $(i = 1, \ldots, N_{\mathrm{sym}})$ are the secret KPS algorithms of S' that are produced similarly to those of S.

Similarly to the key distribution among providers or that between providers and servers, if $G^{(\mu),i_0 j}_{\mathrm{sym}}$ is exposed by a certain attack, S and S' can still share their key using other $Z^{(\mu),i_1 j}_S$ and $Z^{(\mu),i_1 j}_{S'}$. Moreover, concurrent use of their secret KPS algorithms again results in longer keys.

Using $Z_S^{(\mu),i_0 j}$, $Z_S^{(\mu),i_1 j}$ and $Z_{S'}^{(\mu),i_0 j}$, $Z_{S'}^{(\mu),i_1 j}$, the length of the shared key is doubled. Therefore, the keys shared among providers can be at most $N_{\mathrm{sym}}h$ long.

4. EVALUATION AND SECURITY DISCUSSION

4.1. Communications with Hierarchical KPSs

Here we confirm whether or not Hierarchical KPSs can provide the required communication links in practical communication systems. As already discussed, required communication links are consumer–provider, provider–provider (within a group of providers), provider–server (if the group that the provider belongs to and the group that the server belongs to are allowed to communicate with each other) and server–server (within a group of servers). It can be seen that these communications are available by the method described in Sections 3.1–3.4. Hence, it is confirmed that all required functions are provided by Hierarchical KPSs.

Furthermore, a Hierarchical KPS offers a higher level of security than the Matsumoto–Imai scheme. As mentioned in Sections 3.2–3.4, the keys among providers, those between providers and servers and those among servers can be $N_{\mathrm{sym}}h$ bits long, which is more than the length h of keys between consumers and providers. Hence, these communications can be carried out more safely than those by the Matsumoto–Imai scheme, assuming that the number h of matrices for the KPS-center algorithm is the same in the Hierarchical KPS and in the Matsumoto–Imai scheme.

4.2. Collusion attack against $G^{(\mu)}$

In hierarchical KPSs, in order to break the whole system, adversaries have to obtain $G^{(\mu)}$ by using information that they can access. There are mainly three kinds of collusion attacks against $G^{(\mu)}$: the consumers' collusion, the providers' collusion and the consumers and servers' mixed collusion. The servers cannot reveal $G^{(\mu)}$ by themselves.

First, we address collusion attacks by consumers and providers. In order to break the whole system, collusion of m consumers or n providers is necessary from the information theory point of view, due to the fact that the quantity of the center's secret information is hmn bits, whilst the consumer's secret KPS algorithm has hn bits of information and the provider's secret KPS algorithm has hm bits of information.

It should be noted that the mixed collusion between consumers and providers is inefficient since the information available to the consumers and the providers is not independent. The number of either consumers or providers joining in the collusion attack must exceed the corresponding threshold m or n to succeed the attack. The security of the collusion attack of consumers and providers can basically be addressed by the following theorem.

THEOREM 1. *Let $\{C_1, \ldots, C_{t_1}\}$ and $\{P_1, \ldots, P_{t_2}\}$ be the sets of t_1 consumers and t_2 providers, respectively. Let x_{C_i} and $X_{C_i}^{(\mu)}$ be the effective ID of C_i and the secret KPS algorithm of C_i $(i = 1, \ldots, t_1)$, respectively. Let y_{P_j} and $Y_{P_j}^{(\mu)}$ be the effective ID of P_j and the secret KPS algorithm of P_j $(j = 1, \ldots, t_2)$, respectively. Then, in order to uniquely specify $G^{(\mu)}$ by using $X_{C_i}^{(\mu)}$ $(i = 1, \ldots, t_1)$ and $Y_{P_j}^{(\mu)}$ $(j = 1, \ldots, t_2)$, $t_1 \geq m$ or $t_2 \geq n$ is necessary.*

Proof. It is sufficient to prove that there exists more than one different matrix M such that $x_{C_i} \cdot M = 0$ for any x_{C_i} $(1 \leq i \leq t_1)$ and $M \cdot {}^{\mathrm{t}}y_{P_j} = 0$ for any y_{P_j} $(1 \leq j \leq t_2)$, assuming that $t_1 < m$ and $t_2 < n$.

Here, we define a vector space $\mathcal{X}$ and its orthogonal complement $\mathcal{X}^{\perp}$ as follows:

$$\mathcal{X} := \langle x_{C_1}, x_{C_2}, \ldots, x_{C_{t_1}} \rangle \subset GF(2)^m,$$
$$\mathcal{X}^{\perp} := \{x \in GF(2)^m \mid x \cdot {}^{\mathrm{t}}x' = 0 \text{ for all } x' \in \mathcal{X}\}.$$

Then, we have

$$\dim \mathcal{X} = t_x, \quad 1 \leq t_x \leq t_1 \ (\leq m-1),$$
$$\dim \mathcal{X}^{\perp} = m - t_x.$$

This means that $\mathcal{X}^{\perp}$ has a basis consisting of $m - t_x$ linearly independent vectors:

$$\mathcal{X}^{\perp} = \langle v_1, v_2, \ldots, v_{m-t_x} \rangle,$$

where each v_i $(1 \leq i \leq m - t_x)$ is an m-dimensional row vector over $GF(2)$. Hence,

$$M = \left(\sum_{i=1}^{m-t_x} \lambda_{1,i}\, {}^{\mathrm{t}}v_i, \sum_{i=1}^{m-t_x} \lambda_{2,i}\, {}^{\mathrm{t}}v_i, \ldots, \sum_{i=1}^{m-t_x} \lambda_{n,i}\, {}^{\mathrm{t}}v_i \right)$$
$$= ({}^{\mathrm{t}}v_1, {}^{\mathrm{t}}v_2, \ldots, {}^{\mathrm{t}}v_{m-t_x})\Lambda$$

where

$$\Lambda = \begin{pmatrix} \lambda_{1,1} & \lambda_{2,1} & \cdots & \lambda_{n,1} \\ \lambda_{1,2} & \lambda_{2,2} & \cdots & \lambda_{n,2} \\ \vdots & \vdots & \cdots & \vdots \\ \lambda_{1,m-t_x} & \lambda_{2,m-t_x} & \cdots & \lambda_{n,m-t_x} \end{pmatrix},$$

and $\lambda_{i,j}$ $(1 \leq i \leq n,\ 1 \leq j \leq m - t_x) \in GF(2)$. Since $M \cdot {}^{\mathrm{t}}y_{P_j} = 0$ for any y_{P_j} $(1 \leq j \leq t_2)$, we have

$$({}^{\mathrm{t}}v_1, {}^{\mathrm{t}}v_2, \ldots, {}^{\mathrm{t}}v_{m-t_x})\Lambda\, {}^{\mathrm{t}}y_{P_j} = 0$$

for any y_{P_j} $(1 \leq j \leq t_2)$. Therefore,

$$\Lambda\, {}^{\mathrm{t}}y_{P_j} = 0 \tag{1}$$

since $v_1, v_2, \ldots, v_{m-t_x}$ are linearly independent.

Here, we define a vector space $\mathcal{Y}$ and its orthogonal complement $\mathcal{Y}^{\perp}$ as follows:

$$\mathcal{Y} := \langle y_{P_1}, y_{P_2}, \ldots, y_{P_{t_2}} \rangle \subset GF(2)^n,$$
$$\mathcal{Y}^{\perp} := \{y \in GF(2)^n \mid y \cdot {}^{\mathrm{t}}y' = 0 \text{ for all } y' \in \mathcal{Y}\}.$$

Then, we have

$$\dim \mathcal{Y} = t_y, \quad 1 \leq t_y \leq t_2 \ (\leq n-1),$$
$$\dim \mathcal{Y}^{\perp} = n - t_y.$$

This means that $\mathcal{Y}^{\perp}$ has a basis consisting of $n - t_y$ linearly independent vectors:

$$\mathcal{Y}^{\perp} = \langle u_1, u_2, \ldots, u_{n-t_y} \rangle,$$

where each u_i $(1 \leq i \leq n - t_y)$ is an n-dimensional row vector over $GF(2)$. Hence, from Equation (1)

$$\Lambda = \begin{pmatrix} \sum_{i=1}^{n-t_y} \omega_{1,i} u_i \\ \sum_{i=1}^{n-t_y} \omega_{2,i} u_i \\ \vdots \\ \sum_{i=1}^{n-t_y} \omega_{m-t_x,i} u_i \end{pmatrix} = \Omega \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-t_y} \end{pmatrix}$$

where

$$\Omega = \begin{pmatrix} \omega_{1,1} & \omega_{2,1} & \ldots & \omega_{m-t_x,1} \\ \omega_{1,2} & \omega_{2,2} & \ldots & \omega_{m-t_x,2} \\ \vdots & \vdots & \ldots & \vdots \\ \omega_{1,n-t_y} & \omega_{2,n-t_y} & \ldots & \omega_{m-t_x,n-t_y} \end{pmatrix},$$

and $\omega_{i,j}$ $(1 \leq i \leq m - t_x,\ 1 \leq j \leq n - t_y) \in GF(2)$. Therefore, we have

$$M = ({}^{\mathrm{t}}v_1, {}^{\mathrm{t}}v_2, \ldots, {}^{\mathrm{t}}v_{m-t_x})\Omega \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_{n-t_y} \end{pmatrix}.$$

Hence, there exist $2^{(m-t_x)(n-t_y)}$ different M with the choices of $\omega_{i,j}$ $(1 \leq i \leq m - t_x,\ 1 \leq j \leq n - t_y)$. □

Actually, since symmetric matrices are embedded in $G^{(\mu)}$, leakage of one $k_{\mathrm{sel}}^{(\mu),ij}$ brings an $(n-1)/2$ reduction of the collusion threshold of consumers. However, although all $k_{\mathrm{sel}}^{(\mu),ij}$s are exposed, the collusion threshold is still high enough because we assumed $m \gg n$ (note that the collusion threshold of providers cannot be reduced).

Next, we consider the mixed collusion attack by consumers and servers. Although servers and consumers can collude to reveal $G^{(\mu)}$, the influence that the servers can make in the attack is limited. If the selection of rows belonging to embedded symmetric matrices in $G^{(\mu)}$ is somehow exposed, the influence made by the servers could be stronger. However, even in this case, the collusion attack is not effective if the collusion threshold of consumers is sufficiently high. The security of this kind of attack is addressed by the following theorem.

THEOREM 2. *Let $\{C_1, \ldots, C_{t_1}\}$ and $\{S_1, \ldots, S_{t_2}\}$ be the sets of t_1 consumers and t_2 servers, respectively. Let x_{C_k} and $X_{C_k}^{(\mu)}$ be the effective ID of C_k and the secret KPS algorithm of C_k $(k = 1, \ldots, t_1)$, respectively. Let z_{S_l} and $Z_{S_l}^{(\mu),ij_l}$ $(i = 1, \ldots, N_{\mathrm{sym}})$ be the effective ID of S_l and the secret KPS algorithm of S_l $(l = 1, \ldots, t_2)$, respectively, assuming that S_l belongs to $\mathcal{S}_{j_l}$. Even if $k_{\mathrm{sel}}^{(\mu),ij}$ $(i = 1, \ldots, N_{\mathrm{sym}},\ j = 1, \ldots, N_P)$ are exposed, in order to uniquely specify $G^{(\mu)}$ by using $X_{C_k}^{(\mu)}$ $(k = 1, \ldots, t_1)$ and $Z_{S_l}^{(\mu),ij_l}$ $(l = 1, \ldots, t_2,\ i = 1, \ldots, N_{\mathrm{sym}})$, $t_1 + N_{\mathrm{sym}} t_2 \geq m$ is necessary.*

Proof. Here, we assume among S_l $(l = 1, \ldots, t_2)$ that there exist e_i servers which belong to $\mathcal{S}_i$ $(i = 1, \ldots, N_P)$, respectively, such that $\sum_{i=1}^{N_P} e_i = t_2$. Without loss of generality, we also assume that, for $\sum_{i=0}^{j-1} e_i < l < \sum_{i=0}^{j} e_i + 1$, S_l belongs to $\mathcal{S}_j$, where e_0 is defined as zero.

Then, it is sufficient to prove that there exists more than one different matrix M such that for any x_{C_k} $(1 \leq k \leq t_1)$ $x_{C_k} \cdot M = 0$ and $z_{S_l} \cdot D^{ij} = 0$ $(i = 1, \ldots, N_{\mathrm{sym}})$ for any $z_{S_l}(\sum_{b=0}^{j-1} e_b < l < \sum_{b=0}^{j} e_b)$ for $j = 1, \ldots, N_P$, where, similarly to $G_{\mathrm{sym}}^{(\mu),ij}$ in $G^{(\mu)}$, each of D^{ij} $(i = 1, \ldots, N_{\mathrm{sym}},\ j = 1, \ldots, N_P)$ is an embedded symmetric matrix in M, assuming that $t_1 + N_{\mathrm{sym}} t_2 < m$.

Here, we define vector spaces $\mathcal{Z}_j$ $(j = 1, \ldots, N_P)$ and their orthogonal complements $\mathcal{Z}_j{}^{\perp}$ $(j = 1, \ldots, N_P)$ as follows:

$$\mathcal{Z}_j := \left\langle z_{S_l} \left(\sum_{b=0}^{j-1} e_b < l < \sum_{b=0}^{j} e_b + 1 \right) \right\rangle \subset GF(2)^n,$$

$$\mathcal{Z}_j{}^{\perp} := \{ z \in GF(2)^n \mid z \cdot {}^{\mathrm{t}}z' = 0 \text{ for all } z' \in \mathcal{Z}_j \}.$$

Then, we have

$$\dim \mathcal{Z}_j = t_{z_j}, \quad 1 \leq t_{z_j} \leq e_j,$$
$$\dim \mathcal{Z}_j{}^{\perp} = n - t_{z_j}.$$

This means that $\mathcal{Z}_j{}^{\perp}$ has a basis consisting of $n - t_{z_j}$ linearly independent vectors:

$$\mathcal{Z}_j{}^{\perp} = \langle v_1^{(j)}, v_2^{(j)}, \ldots, v_{n-t_{z_j}}^{(j)} \rangle,$$

where each $v_1^{(j)}, v_2^{(j)}, \ldots, v_{n-t_{z_j}}^{(j)}$ is an n-dimensional row vector over $GF(2)$. Therefore, each row (or column) of D^{ij} $(i = 1, \ldots, N_{\mathrm{sym}},\ j = 1, \ldots, N_P)$ can be expressed as a linear combination by using the basis.

Next, we define a vector space $\mathcal{X}$ and its orthogonal complement $\mathcal{X}^{\perp}$ as follows:

$$\mathcal{X} := \langle x_{C_1}, x_{C_2}, \ldots, x_{C_{t_1}} \rangle \subset GF(2)^m,$$

$$\mathcal{X}^{\perp} := \{ x \in GF(q)^m \mid x \cdot {}^{\mathrm{t}}x' = 0 \text{ for all } x' \in \mathcal{X} \}.$$

Then, we have

$$\dim \mathcal{X} = t_x, \quad 1 \leq t_x \leq t_1 \quad (\leq m - 1),$$
$$\dim \mathcal{X}^{\perp} = m - t_x.$$

These results imply that each column of M can be written by a linear combination of a basis v_i $(1 \leq i \leq m')$ for some m' such that $m' \geq m - t_x - N_{\mathrm{sym}} N_P n + \sum_{j=1}^{N_P} (n - t_{z_j}) N_{\mathrm{sym}} (= m - t_x - \sum_{j=1}^{N_P} t_{z_j} N_{\mathrm{sym}} \geq m - t_1 - t_2 N_{\mathrm{sym}} > 0)$, where

TABLE 2. Collusion thresholds to calculate $G^{(\mu)}$, $G_{\mathrm{sym}}^{(\mu),ij}$.

Colluders	$G^{(\mu)}$	$G_{\mathrm{sym}}^{(\mu),ij}$
Providers	n	n
Consumers	m	$m - n + \log_2 n$
Servers	$-$	n†
Providers + servers	n providers	n‡

† Collusion by servers that belong to the group of servers $\mathcal{S}_j$.
‡ Collusion by any providers and servers that belong to $\mathcal{S}_j$.

v_i $(1 \leq i \leq m')$ are linearly independent m-dimensional vectors over $GF(2)$. Namely,

$$M = \left(\sum_{i=1}^{m'} \psi_{1,i}\, {}^{\mathrm{t}}v_i, \sum_{i=1}^{m'} \psi_{2,i}\, {}^{\mathrm{t}}v_i, \ldots, \sum_{i=1}^{m'} \psi_{n,i}\, {}^{\mathrm{t}}v_i \right),$$
$$= ({}^{\mathrm{t}}v_1, {}^{\mathrm{t}}v_2, \ldots, {}^{\mathrm{t}}v_{m'})\Psi$$

where

$$\Psi = \begin{pmatrix} \psi_{1,1} & \psi_{2,1} & \cdots & \psi_{n,1} \\ \psi_{1,2} & \psi_{2,2} & \cdots & \psi_{n,2} \\ \vdots & \vdots & \cdots & \vdots \\ \psi_{1,m'} & \psi_{2,m'} & \cdots & \psi_{n,m'} \end{pmatrix},$$

and $\psi_{i,j}$ $(1 \leq i \leq n,\ 1 \leq j \leq m') \in GF(2)$.

Hence, there exist $2^{m'n}$ different M with the choices of $\psi_{i,j}$ $(1 \leq i \leq m',\ 1 \leq j \leq n)$. □

Theorem 2 implies that, in the worst case, a collusion of N_{s} servers and N_{c} consumers might reveal $G^{(\mu)}$ if $N_{\mathrm{s}}N_{\mathrm{sym}} + N_{\mathrm{c}} \geq m$. Therefore, the proposed scheme is considered secure if m is determined to be sufficiently large. Since for typical security parameter settings we have $N_{\mathrm{c}} \gg N_{\mathrm{s}}N_{\mathrm{sym}}$, this attack is not effective if the collusion threshold of consumers is large enough to prevent only consumers' collusions. Furthermore, the above attack can be performed in the case where the selection of rows belonging to embedded symmetric matrices in $G^{(\mu)}$ is exposed to colluders. Also, we need to note that Theorem 2 does not show a sufficient condition for revealing $G^{(\mu)}$ by consumers and servers, but a necessary condition. A more strict security analysis in which the selection of rows belonging to embedded symmetric matrices in $G^{(\mu)}$ is not exposed is an interesting open problem.

In Table 2, collusion thresholds against $G^{(\mu)}$ are shown. This means that a Hierarchical KPS can be designed as shown above based on the collusion thresholds n for consumers and m for providers. In a conventional KPS, however, mixed collusions can also be effective. This is why in conventional KPSs the collusion threshold should be $(n + m)$, so that for the center algorithm $(n+m)\times(n+m)$ matrices are needed in the Matsumoto–Imai scheme. Based on this assumption, the memory requirements of a Hierarchical KPS and a conventional KPS will be compared in the next section.

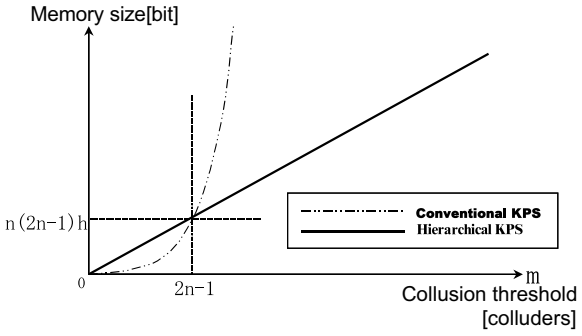

FIGURE 5. Comparison of the required memory size for the KPS-center algorithm in a Hierarchical KPS with that in a conventional KPS, where n indicates the collusion threshold for providers.

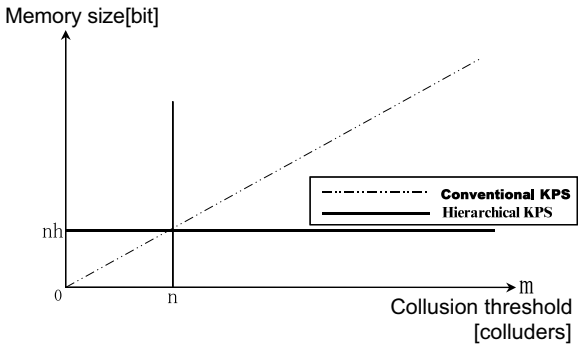

FIGURE 6. Comparison of the required memory size for a consumer's secret KPS algorithm in a Hierarchical KPS with that in a conventional KPS, where n indicates the collusion threshold for providers.

4.3. Memory requirements

Considering these collusion thresholds, m and n are determined mainly by the numbers of consumers and providers, respectively. Similarly, the required memory size for the KPS-center algorithm is determined to be proportional to n times m, while in the Matsumoto–Imai scheme the required memory size for the KPS-center algorithm is proportional to $\frac{1}{2}(n + m)(n + m + 1)$. Furthermore, the memory size for the consumers' secret KPS algorithms is proportional to n. Since in the Matsumoto–Imai scheme this is proportional to $(n + m)$, the memory size for the consumers' secret KPS algorithms can be reduced considerably. Note that for general purpose applications the Matsumoto–Imai scheme, similar to Blundo *et al.*'s [14] and some other schemes, satisfies the memory size optimally of both the KPS center and of the users. Thus, the memory size in the Matsumoto–Imai scheme is regarded as that of a conventional KPS. As the number of consumers is usually much higher than the number of providers, these reductions made in memory size are significant. Figures 5 and 6 show the memory size required for the KPS-center and a consumer. In the Matsumoto–Imai scheme, the required memory size for the KPS center algorithm grows in proportion to the square of the collusion threshold, and the required memory size for users' secret

TABLE 3. Required memory size for each type of entity.

	KPS center	Consumer	Provider	Server
Hierarchical KPS	$hnm + N_P N_{\mathrm{sym}}\lvert k_{\mathrm{sel}}\rvert$	hn	$hm + N_{\mathrm{sym}}\lvert k_{\mathrm{sel}}\rvert$	$hN_{\mathrm{sym}}n$
Conventional KPS	$\frac{1}{2}(n+m)(n+m+1)$	$h(n+m)$	$h(n+m)$	$h(n+m)$

KPS algorithms increases in proportion to the collusion threshold. In contrast, in a Hierarchical KPS, the required memory size for the KPS center algorithm increases in proportion to the collusion threshold for consumers, but the required memory size for consumers' secret KPS algorithms remains unchanged when increasing the collusion threshold for consumers, assuming that the collusion threshold for providers is fixed (since the number of providers is much smaller than the number of consumers, a low collusion threshold will be sufficient to avoid a providers' collusion attack). We also need to note that the KPS center algorithm and providers have to keep the selection of rows in $G^{(\mu)}$ for the embedded symmetric matrices. Hence, $N_P N_{\mathrm{sym}}|k_{\mathrm{sel}}|$-bit and $N_{\mathrm{sym}}|k_{\mathrm{sel}}|$-bit memory will be required additionally for the KPS-center algorithm and for the provider's secret algorithm, respectively, where $|k_{\mathrm{sel}}|$ is the required memory size for a selection of rows belonging to an embedded symmetric matrix in $G^{(\mu)}$.

Also, when taking into account the higher collusion threshold, the difference of the required memory size between a Hierarchical KPS and a conventional KPS will be even more significant.

In summary, the collusion threshold in a Hierarchical KPS can be much higher than that of a conventional KPS when using the same size of memory in the KPS center. Table 3 shows the required memory sizes for each type of entity, assuming that the same collision threshold of consumers is determined in both conventional and Hierarchical KPSs. Only the memory size for the providers is not reduced significantly in a Hierarchical KPS in comparison to that in the Matsumoto–Imai scheme. However, this is not a serious problem since such an amount of memory should be easily available for providers.

4.4. Collusion attack against $G_{\mathrm{sym}}^{(\mu),ij}$

Here, we discuss the collusion attack of consumers against $G_{\mathrm{sym}}^{(\mu),ij}$ in more detail.

Note that, in order to reveal $G_{\mathrm{sym}}^{(\mu),ij}$, the adversary requires n combinations of the consumers' secret KPS algorithms that fulfill the following condition.

CONDITION $(*)$. For the linear sum of the consumers' IDs participating in the combination, all the elements except those selected by $k_{\mathrm{sel}}^{(\mu),ij}$ are entirely zero (see Figure 7).

By using n such combinations, $G_{\mathrm{sym}}^{(\mu),ij}$ can be revealed easily, when the involved sums are linearly independent. Hence, this attack can be realized by only n colluders in the worst case. However, the possibility of its success seems

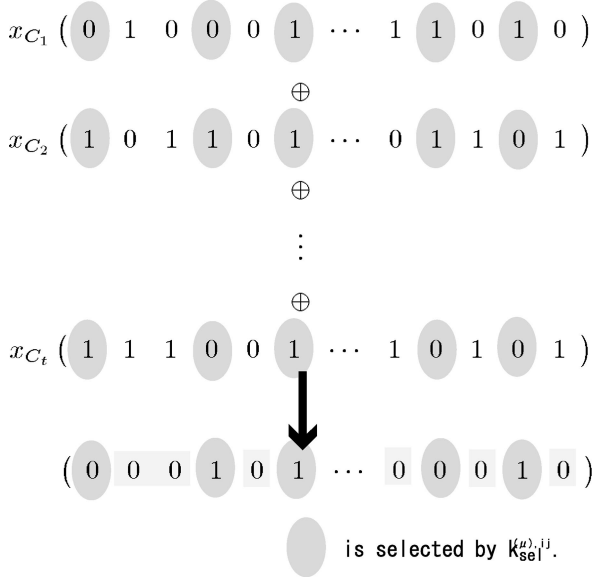

FIGURE 7. The required combination of consumers' IDs to reveal $G_{\mathrm{sym}}^{(\mu),ij}$, where x_{C_i} $(i = 1, \ldots, t)$ are effective identifiers of consumers C_i $(i = 1, \ldots, t)$.

infeasible. Here, we estimate the number of colluders that yields a more feasible possibility to realize the attack.

When t consumers collude, the number of combinations of consumers' IDs is $2^t - 1$. Since the probability that a randomly selected combination fulfills the condition $(*)$ is 2^{n-m}, the expectation $E_{\mathrm{col}}(t)$ of the number of the combinations that fulfill the condition $(*)$ is approximated as follows:

$$E_{\mathrm{col}}(t) = (2^t - 1)(2^{n-m}) \simeq 2^{t+n-m}. \tag{2}$$

Thus, to achieve $E_{\mathrm{col}}(t) \geq n$, we require $t \geq m - n + \log_2 n$. Hence, $m-n+\log_2 n$ can be regarded as the collusion threshold of this attack. Although this threshold seems to be still high, we can find that less than n colluders are required to reveal $G_{\mathrm{sym}}^{(\mu),ij}$ if $k_{\mathrm{sel}}^{(\mu),ij}$ is exposed. Thus, $k_{\mathrm{sel}}^{(\mu),ij}$ must be kept secret from entities with the exception of its legal users. Basically, m and n are defined according to the number of consumers and providers, respectively. However, since the collusion threshold to reveal $G_{\mathrm{sym}}^{(\mu),ij}$ by consumers is defined by both m and n, this must also be considered when choosing m and n. The collusion thresholds against $G_{\mathrm{sym}}^{(\mu),ij}$ are summarized in Table 2. Although $k_{\mathrm{sel}}^{(\mu),ij}$ can be revealed without difficulty if $G^{(\mu)}$ is exposed, we do not need to take care of this attack since the collusion threshold of $G^{(\mu)}$ is set

up high enough to prevent any possible collusion attacks in the real world.

As mentioned in Section 3.2, by embedding multiple symmetric matrices in $G^{(\mu)}$, the damage made by exposing $k_{\mathrm{sel}}^{(\mu),ij}$ can be reduced. Namely, if a $G_{\mathrm{sym}}^{(\mu),ij}$ is revealed, only the group that uses this $G_{\mathrm{sym}}^{(\mu),ij}$ is affected. Even though a $G_{\mathrm{sym}}^{(\mu),ij}$ has been damaged, the communication can still be realized by using another $G_{\mathrm{sym}}^{(\mu),ij}$.

Additionally, although a collusion attack of providers can also reveal $G^{(\mu)}$, the collusion threshold of this attack is the same as that of an attack against $G_{\mathrm{sym}}^{(\mu),ij}$ by the providers. Hence, in order to reveal $G_{\mathrm{sym}}^{(\mu),ij}$, the providers have to reveal $G^{(\mu)}$. Moreover, by a collusion attack of servers, $G_{\mathrm{sym}}^{(\mu),ij}$ can be revealed. However, only the servers that belong to S_j can carry out this attack. In such an attack, the collusion threshold is n, and it is regarded as being high enough because there are not many servers in comparison to the number of consumers.

4.5. Applications

A Hierarchical KPS can be applied to many kinds of communication systems. In practical communication systems, we often find two kinds of entities that are regarded as consumers and providers. Usually, a minority of entities in the system communicate with almost all of the other entities, while the majority communicate only with specific entities (or the minority). Hence, we can regard the minority and the majority as providers and consumers, respectively. Furthermore, in communication systems, we often find entities that provide information to specific providers. Such entities are regarded as servers.

As an example, in broadcasting, addressees and broadcasting stations can be regarded as consumers and providers, respectively. Certain entities that serve information for the broadcasting stations take the role of servers. Assuming that the numbers of addressees, broadcasting stations and servers are 10,000,000, 5000 and 200,000, respectively, we can set up $m = 131{,}072$ and $n = 512$ approximately. Then the number $N_{\mathrm{sym}} \cdot N_P$ of embedded symmetric matrices is 256. Thus, the collusion threshold of addressees is $m = 131{,}072$, which is 16 times as large as the number 8192 with a conventional KPS, assuming that the utilized memory size is the same in both the Hierarchical KPS and the Matsumoto–Imai scheme. In this case, for the Matsumoto–Imai scheme 8192×8192 symmetric matrices are used as the KPS-center algorithm. Even when all of the information for the location of embedded symmetric matrices in the center algorithm is exposed, the collusion threshold is still eight times that of a conventional KPS. Furthermore, the memory requirement (using $h = 64$ bits) is $hn = 32{,}768$ bits ($= 4$ kbytes), which is $1/16$ of the requirement of 64 kbytes required in a conventional KPS.

An electronic toll collection (ETC) system may also be one of the applications of our scheme. We have shown an efficient credit-payment system for ETC and an optimized KPS for it in [19]. This optimized KPS can be regarded as a particular implementation of a Hierarchical KPS.

4.6. Generalization of Hierarchical KPSs

It is a well known fact that for most KPSs the center algorithm can be described as symmetric matrices [9, 10, 11, 12, 14]. Therefore, by replacing these matrices with the particular asymmetric matrices described in this paper (see Figure 3) it is possible to construct the hierarchical structure (see Figure 1) based on these KPSs. For example, in a straightforward manner, we can construct a hierarchical KPS based on Blom's scheme [9]. The performance of this scheme is exactly the same as that of the Hierarchical KPS based on the Matsumoto–Imai scheme.

5. CONCLUSION

In this paper, a Hierarchical KPS, which is a new style of KPS, was proposed. It has been pointed out that certain collusion attacks can be effective against KPSs. On the other hand, it has been shown how KPSs can be improved for practical communication systems to increase their resistance against collusion attacks. To be specific, by removing communication links that are not required in a practical communication system, resistance against collusion attacks is increased significantly. For a typical security parameter setting, the collusion threshold of the improved KPS is 16 times higher than that of the conventional KPS while using the same amount of memory in the KPS center. The memory required by the users is even reduced to be $1/16$ of that for the conventional KPS. Hence, a Hierarchical KPS provides a higher level of security against collusion attacks and offers a simplified implementation due to its reduced memory sizes. This makes a Hierarchical KPS attractive for various applications like broadcasting and E-commerce on the Internet. Additionally, since public-key cryptosystems do not have great advantages over KPSs in terms of computational cost, ID-basedness and so on, the efficient combination of a public-key cryptosystem and our scheme will produce a more efficient and secure communication system than one single use of public-key cryptosystems.

ACKNOWLEDGEMENTS

Part of this work was performed as part of a Research for the Future Program (RFTP) supported by the Japanese Society for the Promotion of Science (JSPS) under contract No. JSPS-RFTF 96P00604. The first author is supported by a Research Fellowship of the JSPS. Part of this research was presented at ASIACRYPT'99 [20].

REFERENCES

[1] Fiat, A. and Shamir, A. (1986) How to prove yourself: practical solutions to identification and signature problems. In *Proc. CRYPTO'86. Lecture Notes in Computer Science*, **263**, 186–194. Springer, Berlin.

[2] Shamir, A. (1985) Identity-based cryptosystems and signature schemes. In *Advances in Cryptology–CRYPTO'84. Lecture Notes in Computer Science*, **196**, 47–53. Springer, Berlin.

[3] Maurer, U. and Yacobi, Y. (1992) Non-interactive public-key cryptography. In *Advances in Cryptology—EUROCRYPT'91. Lecture Notes in Computer Science*, **547**, 498–507. Springer, Berlin.

[4] Maurer, U. and Yacobi, Y. (1993) A remark on a non-interactive public-key distribution system. In *Advances in Cryptology—EUROCRYPT'92. Lecture Notes in Computer Science*, **658**, 458–460. Springer, Berlin.

[5] Okamoto, E. and Tanaka, K. (1989) Identity-based information security management system for personal comuputer networks. *IEEE J. Selected Areas Commun.*, **7**, 290–294.

[6] Tanaka, H. (1988) A realization scheme of the identity-based cryptosystems. In *Advances in Cryptology—CRYPTO'87. Lecture Notes in Computer Science*, **293**, 340–349. Springer, Berlin.

[7] Tsujii, S. and Chao, J. (1992) A new ID-based key sharing system. In *Advances in Cryptology–CRYPTO'91. Lecture Notes in Computer Science*, **576**, 288–299. Springer, Berlin.

[8] Coppersmith, D. (1994) Attack on the cryptographic scheme NIKS-TAS. In *Advances in Cryptology—CRYPTO'94. Lecture Notes in Computer Science*, **839**, 40–49. Springer, Berlin.

[9] Blom, R. (1983) Non-public key distribution. *Advances in Cryptology–CRYPTO'82*, pp. 231–236. Plenum, New York.

[10] Matsumoto, T. and Imai, H. (1988) On the KEY PREDISTRIBUTION SYSTEM: a practical solution to the key distribution problem. In *Advances in Cryptology–CRYPTO'87. Lecture Notes in Computer Science*, **293**, 185–193. Springer, Berlin.

[11] Gong, L. and Wheeler, D. J. (1993) A matrix key-distribution scheme. *J. Cryptology*, **2**, 51–59.

[12] Jackson, W. A., Martin, K. M. and O'Keefe, C. M. (1994) Multisecret threshold schemes. In *Advances in Cryptology–CRYPTO'93, Lecture Notes in Computer Science*, **773**, 126–135. Springer, Berlin.

[13] Desmedt, Y. and Viswanathan, V. (1998) Unconditionally secure dynamic conference key distribution. *ISIT'98*, Cambridge, MA, August 16–21. IEEE, New York.

[14] Blundo, C. *et al.* (1993) Perfectly secure key distribution for dynamic conferences. In *Advances in Cryptology—CRYPTO'92. Lecture Notes in Computer Science*, **740**, 471–486. Springer, Berlin.

[15] Fiat, A. and Naor, M. (1994) Broadcast encryption. In *Advances in Cryptology—CRYPTO'93. Lecture Notes in Computer Science*, **773**, 480–491. Springer, Berlin.

[16] Blundo, C., Frota Mattos, L. A. and Stinson, D. R. (1996) Trade-offs between communication and storage in unconditionally secure schemes for broadcast encryption and interactive key distribution. In *Advances in Cryptology—CRYPTO'96. Lecture Notes in Computer Science*, **1109**, 387–400. Springer, Berlin.

[17] Kurosawa, K., Yoshida, T., Desmedt, Y. and Burmester, M. (1998) Some bounds and a construction for secure broadcast encryption. In *Advances in Cryptology–ASIACRYPT'98. Lecture Notes in Computer Science*, **1514**, 420–433. Springer, Berlin.

[18] Matsumoto, T. *et al.* (1990) A prototype KPS and its application—IC card based key sharing and cryptographic communication. *IEICE Trans.*, **E73**, 1111–1119.

[19] Hanaoka, G., Nishioka, T., Zheng, Y. and Imai, H. (2000) An optimization of credit-based payment for electronic toll collection systems. *IEICE Trans.*, **E83-A**, 1681–1690.

[20] Hanaoka, G., Nishioka, T., Zheng, Y. and Imai, H. (1999) An efficient hierarchical identity-based key-sharing method resistant against collusion-attacks. In *Advances in Cryptology–ASIACRYPT'99. Lecture Notes in Computer Science*, **1716**, 348–362. Springer, Berlin.