PAPER *Special Section on Cryptography and Information Security*

Optimal Multiple Assignments Based on Integer Programming in Secret Sharing Schemes with General Access Structures*

Mitsugu IWAMOTO†a), *Member*, **Hirosuke YAMAMOTO**††, *Fellow*, *and* **Hirohisa OGAWA**†††, *Nonmember*

SUMMARY It is known that for any general access structure, a secret sharing scheme (SSS) can be constructed from an (m, m)-threshold scheme by using the so-called *cumulative map* or from a (t, m)-threshold SSS by a modified cumulative map. However, such constructed SSSs are not efficient generally. In this paper, a new method is proposed to construct a SSS from a (t, m)-threshold scheme for any given general access structure. In the proposed method, integer programming is used to derive the optimal (t, m)-threshold scheme and the optimal distribution of the shares to minimize the average or maximum size of the distributed shares to participants. From the optimality, it can always attain lower coding rate than the cumulative maps because the cumulative maps cannot attain the optimal distribution in many cases. The same method is also applied to construct SSSs for incomplete access structures and/or ramp access structures.

key words: *secret sharing schemes, general access structures, multiple assignment map, integer programming, ramp schemes*

1. Introduction

A Secret Sharing Scheme [1], [2] (SSS) is a method to share a secret information S among a set of participants securely. The secret S is encoded to so-called *shares* $V_1, V_2, \ldots, V_n$, each of which is distributed to a participant. In the case (k, n)-threshold SSSs, the secret S can be reproduced from any k out of n shares although any $k-1$ or less shares do not leak any information of S. Hence, S is secure even if $k-1$ shares are stolen or $n-k$ shares are destroyed. The (k, n)-threshold access structure can be generalized to so-called *general access structures* which consist of the families of *qualified sets* and *forbidden sets*. A qualified set is the subset of shares that can reproduce the secret, but a forbidden set is the subset that does not leak out any information of S.

Generally, the efficiency of a SSS is measured by the entropy of each share. It is known that for any access structures, the entropies of secret S and shares V_i, $i = 1, 2, \ldots, n$, must satisfy $H(V_i) \geq H(S)$ [3]–[5]. Furthermore, in the case of (k, n)-threshold SSSs, the optimal SSSs attaining $H(V_i) = H(S)$ can easily be constructed [1]. However, it is hard to derive efficient SSSs for arbitrarily given general access structures although several construction methods have been proposed.

For example, the *monotone circuit construction* [6] is a method to realize a SSS by combining several (m, m)-threshold SSSs. This method is simple but inefficient, and hence, it is extended to the *decomposition construction* [7], which uses several decomposed general SSSs. Although the decomposition construction can attain the optimal coding rates for some special access structures, it cannot construct an efficient SSS in the case that the decomposed SSSs cannot be realized efficiently. Note that a monotone circuit construction is based on qualified sets. Hence, as another extension of monotone circuit construction, a method is proposed to construct a SSS with general access structures based on qualified sets and (t, m)-threshold SSSs [8].

On the other hand, for any given general access structure, a SSS can be constructed from a (t, m)-threshold SSS by a multiple assignment map such that t or more shares of the (t, m)-threshold SSS are assigned to qualified sets but $t-1$ or less shares are assigned to forbidden sets. The *cumulative map* is a simple realization of the multiple assignment map based on an (m, m)-threshold SSS [9]–[11], and from the simplicity, it is often used in visual secret sharing schemes for general access structures [12], [13]. However, it is known that the SSS constructed by the cumulative map is inefficient generally, especially in the case that the access structure is a (k, n)-threshold SSS with $k \neq n$. Recently, a *modified* cumulative map based on a (t, m)-threshold SSS is proposed to overcome this defect [14]. But, the modified cumulative map is not always more efficient than the original cumulative map.

In this paper, we propose a new construction method that can derive the optimal multiple assignment map by integer programming. The proposed construction method is simple and optimal in the sense of multiple assignment maps. Furthermore, it can also be applied to incomplete and/or ramp access structures.

This paper is organized as follows. In Sect. 2, we give the definitions of SSSs and introduce the multiple assignment map. We also introduce the construction methods of the cumulative map and the modified cumulative map, and we point out their defects. To overcome such defects, we propose a new construction method of the optimal multiple assignment map by integer programming in Sect. 3. Furthermore, Sections 4 and 5 are devoted to present the appli-

Manuscript received March 23, 2006.
Manuscript revised June 28, 2006.
Final manuscript received August 30, 2006.

†The author is with Graduate School of Information Systems, University of Electro-Communications, Chofu-shi, 182-8585 Japan.

††The author is with Graduate School of Frontier Sciences, University of Tokyo, Kashiwa-shi, 277-8561 Japan.

†††The author is with C4 technology, Inc., Tokyo, 141-0021 Japan.

*This paper was presented at the technical meeting on ISEC of IEICE, 2003, and the IEEE International Symposium on Information Theory, 2004.

a) E-mail: mitsugu@hn.is.uec.ac.jp

DOI: 10.1093/ietfec/e90–a.1.101

cations of the proposed method to incomplete or ramp SSSs for general access structures, respectively.

2. Preliminaries

2.1 Definitions

Throughout this paper, a set of shares and a family of share sets are represented by bold-face and script letters, respectively. For sets $\boldsymbol{A}$ and $\boldsymbol{B}$, we denote a difference set by $\boldsymbol{A}-\boldsymbol{B}$, which is defined as $\boldsymbol{A} - \boldsymbol{B} \stackrel{\text{def}}{=} \boldsymbol{A} \cap \overline{\boldsymbol{B}}$ where $\overline{\boldsymbol{B}}$ means the complement of $\boldsymbol{B}$. Furthermore, the cardinality of $\boldsymbol{A}$ is represented by $|\boldsymbol{A}|$, and the Cartesian product of $\boldsymbol{A}$ and $\boldsymbol{B}$ is expressed by $\boldsymbol{A} \times \boldsymbol{B}$.

Let $\boldsymbol{V} = \{V_1, V_2, \ldots, V_n\}$ be the entire set of shares, and let $2^{\boldsymbol{V}}$ be the entire family of subsets of $\boldsymbol{V}$. We represent the family of qualified sets that can reproduce a secret information S and the family of forbidden sets that cannot gain any information of S by $\mathcal{A}_1$ and $\mathcal{A}_0$, respectively.

$\Gamma = \{\mathcal{A}_1, \mathcal{A}_0\}$ is called an *access structure*. For instance, the access structure of (k, n)-threshold SSSs can be represented as follows:

$$\mathcal{A}_1 = \{\boldsymbol{A} \in 2^{\boldsymbol{V}} : k \leq |\boldsymbol{A}| \leq n\}, \tag{1}$$

$$\mathcal{A}_0 = \{\boldsymbol{A} \in 2^{\boldsymbol{V}} : 0 \leq |\boldsymbol{A}| \leq k-1\}. \tag{2}$$

In SSSs, it obviously holds that $\mathcal{A}_1 \cap \mathcal{A}_0 = \emptyset$. If it also holds that $\mathcal{A}_1 \cup \mathcal{A}_0 = 2^{\boldsymbol{V}}$, the access structure is called *complete*. If $\mathcal{A}_1 \cup \mathcal{A}_0 \neq 2^{\boldsymbol{V}}$, it is called *incomplete*. Note that any access structure must satisfy the following *monotonicity*.

$$\boldsymbol{A} \in \mathcal{A}_1 \;\Rightarrow\; \boldsymbol{A}' \in \mathcal{A}_1 \text{ for all } \boldsymbol{A}' \supseteq \boldsymbol{A} \tag{3}$$

$$\boldsymbol{A} \in \mathcal{A}_0 \;\Rightarrow\; \boldsymbol{A}' \in \mathcal{A}_0 \text{ for all } \boldsymbol{A}' \subseteq \boldsymbol{A} \tag{4}$$

Therefore, we can define the family of *minimal* qualified sets and the family of *maximal* forbidden sets as follows:

$$\mathcal{A}_1^- = \{\boldsymbol{A} \in \mathcal{A}_1 : \boldsymbol{A} - \{V\} \notin \mathcal{A}_1 \text{ for any } V \in \boldsymbol{A}\}, \tag{5}$$

$$\mathcal{A}_0^+ = \{\boldsymbol{A} \in \mathcal{A}_0 : \boldsymbol{A} \cup \{V\} \notin \mathcal{A}_0 \text{ for any } V \in \boldsymbol{V} - \boldsymbol{A}\}. \tag{6}$$

We assume that the secret information S and each share V_i are random variables, which take values in finite fields $\mathbb{F}_S$ and $\mathbb{F}_{V_i}$, respectively. Then, share set $\boldsymbol{A} = \{V_{i_1}, V_{i_2}, \ldots, V_{i_u}\} (\subseteq \boldsymbol{V})$, which takes values in $\mathbb{F}_{\boldsymbol{A}} \stackrel{\text{def}}{=} \mathbb{F}_{V_{i_1}} \times \mathbb{F}_{V_{i_2}} \times \cdots \times \mathbb{F}_{V_{i_u}}$, must satisfy the following conditions:

$$H(S|\boldsymbol{A}) = H(S) \text{ if } \boldsymbol{A} \in \mathcal{A}_0, \tag{7}$$

$$H(S|\boldsymbol{A}) = 0 \quad \text{ if } \boldsymbol{A} \in \mathcal{A}_1, \tag{8}$$

where $H(S)$ is the entropy of S and $H(S|\boldsymbol{A})$ is the conditional entropy of S for a given $\boldsymbol{A}$.

Now, let us define the coding rate of a share V_i as $\rho_i \stackrel{\text{def}}{=} H(V_i)/H(S)$, for $i = 1, 2, \ldots, n$[†]. Since each ρ_i may be different in the case of general access structures, it is cumbersome to treat each ρ_i individually. Hence, we consider only the following *average* coding rate $\tilde{\rho}$ and *worst* coding rate ρ^*.

$$\tilde{\rho} \stackrel{\text{def}}{=} \frac{1}{n} \sum_{i=1}^{n} \rho_i, \tag{9}$$

$$\rho^* \stackrel{\text{def}}{=} \max_{1 \leq i \leq n} \rho_i. \tag{10}$$

For a given access structure $\Gamma = \{\mathcal{A}_1, \mathcal{A}_0\}$, we call $V \in \boldsymbol{V}$ a *significant* share if there exists a share set $\boldsymbol{A} \in 2^{\boldsymbol{V}}$ such that $\boldsymbol{A} \cup \{V\} \in \mathcal{A}_1$ but $\boldsymbol{A} \in \mathcal{A}_0$.

Remark 1: Note that a non-significant share plays no roll in the SSS, and hence, $\rho_i = 0$ can always be attained for each non-significant share V_i in any access structure Γ. Furthermore, if there exists a non-significant share V_i with $\rho_i > 0$, the average coding rate can be reduced by setting $\rho_i = 0$ without changing all the significant shares. Hence, we call a non-significant share a *vacuous* share. On the other hand, we have $\rho_i \geq 1$ for any significant share V_i because it must satisfy $H(V_i) \geq H(S)$ [3]–[5]. In the following, we assume that every share is significant. □

If a SSS attains $\rho_i = 1$ for all i, it is called *ideal*. It is known that in the case of (k, n)-threshold SSSs, the ideal SSS can easily be constructed for any k and n [1]. Since $\rho_i \geq 1$, $i = 1, 2, \ldots, n$, must hold for any significant share V_i in any access structures, $\tilde{\rho} = 1$ or $\rho^* = 1$ are the necessary and sufficient conditions for a SSS to be ideal [4].

2.2 Multiple Assignment Map

In this paper, we consider the construction of SSSs with general access structures by distributing one or more shares of a (t, m)-threshold SSS to each participant. The construction method can generally be formulated by the *multiple assignment map* [9]–[11], which is explained in the following.

Let $\Gamma = \{\mathcal{A}_1, \mathcal{A}_0\}$ be a given general access structure with the entire share set $\boldsymbol{V} = \{V_1, V_2, \ldots, V_n\}$ and let $\boldsymbol{W}_{(t,m)} = \{W_1^{(t)}, W_2^{(t)}, \ldots, W_m^{(t)}\}$ be the entire share set of a (t, m)-threshold SSS. We now consider a map $\varphi_\Gamma : \{1, 2, \ldots, n\} \to 2^{\boldsymbol{W}_{(t,m)}}$, which assigns each $i \in \{1, 2, \ldots, n\}$ a subset of shares generated by the (t, m)-threshold scheme, and a map $\Phi_\Gamma : 2^{\boldsymbol{V}} \to 2^{\boldsymbol{W}_{(t,m)}}$, which is defined as $\Phi_\Gamma(\boldsymbol{A}) \stackrel{\text{def}}{=} \bigcup_{V_i \in \boldsymbol{A}} \varphi_\Gamma(i)$ for a share set $\boldsymbol{A} \subseteq \boldsymbol{V}$. Then, φ_Γ is called a *multiple assignment map* for the access structure Γ if $\Phi_\Gamma(\boldsymbol{A})$ satisfies the following conditions [9]–[11]:

$$|\Phi_\Gamma(\boldsymbol{A})| \geq t \qquad \text{if } \boldsymbol{A} \in \mathcal{A}_1, \tag{11}$$

$$|\Phi_\Gamma(\boldsymbol{A})| \leq t-1 \quad \text{if } \boldsymbol{A} \in \mathcal{A}_0, \tag{12}$$

$$\Phi_\Gamma(\boldsymbol{V}) = \boldsymbol{W}_{(t,m)}. \tag{13}$$

It is easy to see that any SSS with access structure Γ satisfying (7) and (8) can be realized by letting $V_i = \varphi_\Gamma(i)$ if

[†] The coding rate ρ_i implies the number of bits of V_i per one bit of the secret S. We note that a coding rate is often defined as $H(S)/H(V_i)$, which is the inverse of ρ_i, since it takes a value between zero and one. However, $H(S)/H(V_i)$ can become larger than one in ramp SSSs. See Sect. 5. We use ρ_i as the coding rate in this paper because ramp SSSs are treated in addition to normal SSSs and the average coding rate $\tilde{\rho}$ can be simply defined as (9).

(11)–(13) are satisfied. To distinguish $W_j^{(t)} \in \boldsymbol{W}_{(t,m)}$ from the shares V_i of Γ, we call $W_j^{(t)}$ a *primitive* share.

Remark 2: In the multiple assignment maps, only *one* (t,m)-threshold SSS is used. Although SSSs with general access structures can be constructed by two or more threshold SSSs, e.g., [6], [8], [14], [14, proposed scheme II], and [15], they are out of the scope of this paper. □

Since any (t,m)-threshold SSS can easily be constructed as an ideal SSS [1], [3], we assume in this paper that the (t,m)-threshold SSS with $\boldsymbol{W}_{(t,m)} = \{W_1^{(t)}, W_2^{(t)}, \ldots, W_m^{(t)}\}$ is ideal. Then, the average and worst coding rates defined by (9) and (10) become

$$\tilde{\rho} = \frac{1}{n}\sum_{i=1}^{n} |\varphi_\Gamma(i)|, \tag{14}$$

$$\rho^* = \max_{1\le i\le n} |\varphi_\Gamma(i)|, \tag{15}$$

respectively, since it holds that $H(W_j^{(t)}) = H(S)$ and, hence, $\rho_i = |\varphi_\Gamma(i)|$.

In the case of $t = m$, it is known that the multiple assignment map φ_Γ satisfying (11)–(13) can be realized for any access structures [9]–[11]. Suppose that the access structure $\Gamma = \{\mathcal{A}_1, \mathcal{A}_0\}$ has

$$\mathcal{A}_0^+ = \{\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_m\}. \tag{16}$$

Since $m = |\mathcal{A}_0^+|$, consider the map $\psi_\Gamma : \{1,2,\ldots,n\} \to 2^{\boldsymbol{W}_{(m,m)}}$ defined by

$$\psi_\Gamma(i) = \bigcup_{j:V_i\notin \boldsymbol{F}_j} \left\{W_j^{(m)}\right\} \tag{17}$$

where $\boldsymbol{F}_j \in \mathcal{A}_0^+$ and $\boldsymbol{W}_{(m,m)} = \{W_1^{(m)}, W_2^{(m)}, \ldots, W_m^{(m)}\}$ is the entire set of primitive shares of an (m,m)-threshold SSS. The above multiple assignment map ψ_Γ is called the *cumulative map* [9]–[11], [16].

Example 3: Assume that $n = 4$ and access structure Γ_1 is defined by

$$\mathcal{A}_1^- = \{\{V_1,V_2,V_3\},\{V_1,V_4\},\{V_2,V_4\},\{V_3,V_4\}\}, \tag{18}$$

$$\mathcal{A}_0^+ = \{\{V_1,V_2\},\{V_1,V_3\},\{V_2,V_3\},\{V_4\}\}. \tag{19}$$

Then, $m = \left|\mathcal{A}_0^+\right| = 4$, and the cumulative map ψ_{Γ_1} is given from (17) as follows.

$$V_1 = \psi_{\Gamma_1}(1) = \left\{W_3^{(4)}, W_4^{(4)}\right\}, \tag{20}$$

$$V_2 = \psi_{\Gamma_1}(2) = \left\{W_2^{(4)}, W_4^{(4)}\right\}, \tag{21}$$

$$V_3 = \psi_{\Gamma_1}(3) = \left\{W_1^{(4)}, W_4^{(4)}\right\}, \tag{22}$$

$$V_4 = \psi_{\Gamma_1}(4) = \left\{W_1^{(4)}, W_2^{(4)}, W_3^{(4)}\right\}. \tag{23}$$

In this example, it holds that $\tilde{\rho} = 9/4$ and $\rho^* = 3$. □

It is known that the next theorem holds for the cumulative map ψ_Γ.

Theorem 4 ([16]): For any multiple assignment map $\varphi_\Gamma : \{1,2,\ldots,n\} \to 2^{\boldsymbol{W}_{(t,m)}}$ with $t = m$, it must hold that $|\boldsymbol{W}_{(m,m)}| \ge |\mathcal{A}_0^+|$, i.e., $m \ge |\mathcal{A}_0^+|$. The equality holds if and only if $\varphi_\Gamma(i)$ is equal to the cumulative map $\psi_\Gamma(i)$ defined by (17) except the freedom of permutations of $\boldsymbol{F}_j$'s in (16). □

Theorem 4 means that, in the case of $t = m$, the cumulative map ψ_Γ minimizes the number of primitive shares m. But, the minimization of m does not mean the realization of an efficient SSS generally because it does not minimize the average coding rate $\tilde{\rho}$ and/or the worst coding rate ρ^*.

For instance, consider the case that Γ is a (k,n)-threshold access structure with $k \neq n$. If we construct shares V_i by the cumulative map ψ for this Γ, each V_i must consist of $\binom{n-1}{k-1}$ primitive shares of an $\left(\binom{n}{k-1},\binom{n}{k-1}\right)$-threshold SSS because of $|\mathcal{A}_0^+| = \binom{n}{k-1}$. This means that $\tilde{\rho} = \rho^* = \binom{n-1}{k-1}$. But, if we use the (k,n)-threshold SSS itself, we have $\tilde{\rho} = \rho^* = 1$ because each V_i consists of one primitive share. Hence, the cumulative map is quite inefficient in the case that Γ is a (k,n)-threshold access structure. In order to overcome this defect, a *modified* cumulative map is proposed in [14] based on (t,m)-threshold SSSs†. The modified cumulative map ψ'_Γ is constructed as follows.

Construction 5 ([14]): For a given $\Gamma = \{\mathcal{A}_0, \mathcal{A}_1\}$ and a positive integer $g \overset{\text{def}}{=} \min_{\boldsymbol{A}\in\mathcal{A}_1^-} |\boldsymbol{A}|$, let $\mathcal{G}_0 \subseteq \mathcal{A}_0^+$ be the family defined by

$$\mathcal{G}_0 = \{\boldsymbol{G} \in \mathcal{A}_0^+ : |\boldsymbol{G}| \ge g\}. \tag{24}$$

When $\mathcal{G}_0 = \{\boldsymbol{G}_1, \boldsymbol{G}_2, \ldots, \boldsymbol{G}_u\} \neq \emptyset$, let $l_j \overset{\text{def}}{=} |\boldsymbol{G}_j| - g + 1$ for $j = 1,2,\ldots,u$, and $\ell_j \overset{\text{def}}{=} \sum_{p=1}^{j} l_p$. If $\mathcal{G}_0 = \emptyset$, let $u = 1$ and $\ell_1 = 0$. Then, consider a $(g+\ell_u, n+\ell_u)$-threshold SSS and the set of primitive shares $\boldsymbol{W}_{(g+\ell_u, n+\ell_u)} = \{W_1^{(g+\ell_u)}, W_2^{(g+\ell_u)}, \ldots, W_{n+\ell_u}^{(g+\ell_u)}\}$. Furthermore, let $\boldsymbol{U}_j$, $j = 1,2,\ldots,u$, be the subset of primitive shares defined by

$$\boldsymbol{U}_1 = \emptyset \quad \text{if} \quad \mathcal{G}_0 = \emptyset, \tag{25}$$

$$\boldsymbol{U}_j = \left\{W_{n+\ell_{j-1}+1}^{(g+\ell_u)}, W_{n+\ell_{j-1}+2}^{(g+\ell_u)}, \ldots, W_{n+\ell_j}^{(g+\ell_u)}\right\} \quad \text{if} \quad \mathcal{G}_0 \neq \emptyset, \tag{26}$$

where $\ell_0 = 0$. Then, the modified cumulative map ψ'_Γ is defined by

$$\psi'_\Gamma(i) = \left\{W_i^{(g+\ell_u)}\right\} \cup \left\{\bigcup_{j:V_i\notin \boldsymbol{G}_j} \boldsymbol{U}_j\right\}. \tag{27}$$

□

In the case where Γ is a (k,n)-threshold access structure, it holds that $\mathcal{G}_0 = \emptyset$ and $\boldsymbol{U}_1 = \emptyset$, and hence, it holds that $\psi'_\Gamma(i) = \{W_i^{(k)}\}$ for $i = 1,2,\ldots,n$ and this scheme coincides with the ideal (k,n)-threshold SSS [14]. Therefore,

†Since proposed scheme II in [14] uses two or more (t,m)-threshold SSSs, it is not considered in this paper. See Remark 2.

the modified cumulative map ψ'_Γ is efficient if Γ is, or is similar to, a (k,n)-threshold access structures. Furthermore, it is shown in [14] that if the access structure Γ satisfies

$$\left|\mathcal{A}_0^+\right| \geq \frac{(n-g-1)\ell_u + n + 2|\mathcal{G}_0|}{n-g+1}, \tag{28}$$

then it holds that for the original cumulative map ψ_Γ, $\sum_{V_i \in \boldsymbol{V}} |\psi'_\Gamma(i)| \leq \sum_{V_i \in \boldsymbol{V}} |\psi_\Gamma(i)|$, which means that the average coding rate $\tilde{\rho}$ of ψ'_Γ is smaller than or equal to ψ_Γ.

But, as shown in the following example, ψ'_Γ is not always more efficient than ψ_Γ if Γ does not satisfy (28).

Example 6: Consider the access structure Γ_1 given by (18) and (19) in Example 3, which does not satisfy (28). Since we have $g = 2$ from (18), $\mathcal{G}_0$ becomes $\mathcal{G}_0 = \{\{V_1, V_2\}, \{V_1, V_3\}, \{V_2, V_3\}\} \overset{\text{def}}{=} \{\boldsymbol{G}_1, \boldsymbol{G}_2, \boldsymbol{G}_3\}$. Furthermore, since we have that $l_1 = l_2 = l_3 = 1$ and $\ell_3 = 3$, $\boldsymbol{U}_i$'s are determined as $\boldsymbol{U}_1 = \{W_5^{(5)}\}$, $\boldsymbol{U}_2 = \{W_6^{(5)}\}$, $\boldsymbol{U}_3 = \{W_7^{(5)}\}$ for $\boldsymbol{W}_{(5,7)} = \{W_1^{(5)}, W_2^{(5)}, \ldots, W_7^{(5)}\}$. Hence, we can check that Γ_1 does not satisfy (28) because of $|\mathcal{A}_0^+| = 4$, $n = 4$, $g = 2$, $\ell_u = 3$, and $|\mathcal{G}_0| = 3$. Finally, we have from (27) that

$$V_1 = \psi'_{\Gamma_1}(1) = \left\{W_1^{(5)}, W_7^{(5)}\right\}, \tag{29}$$
$$V_2 = \psi'_{\Gamma_1}(2) = \left\{W_2^{(5)}, W_6^{(5)}\right\}, \tag{30}$$
$$V_3 = \psi'_{\Gamma_1}(3) = \left\{W_3^{(5)}, W_5^{(5)}\right\}, \tag{31}$$
$$V_4 = \psi'_{\Gamma_1}(4) = \left\{W_4^{(5)}, W_5^{(5)}, W_6^{(5)}, W_7^{(5)}\right\}. \tag{32}$$

In this example, the coding rates are given by $\tilde{\rho} = 5/2$ and $\rho^* = 4$, which are larger than the coding rates of Example 3, i.e., $\tilde{\rho} = 9/4$ and $\rho^* = 3$. □

Note that (28) does not guarantee that the worst coding rate ρ^* of ψ'_Γ is smaller than ψ_Γ. Actually, the next example shows a case where ψ'_Γ attains a smaller average coding rate but gives larger worst coding rate than ψ_Γ.

Example 7: Consider the access structure Γ_2 given by

$$\begin{aligned}
\mathcal{A}_1^- = &\{\{V_1, V_2, V_3, V_5\}, \{V_1, V_2, V_4\}, \{V_1, V_3, V_4\},\\
&\{V_1, V_4, V_5\}, \{V_2, V_3, V_4\},\\
&\{V_2, V_4, V_5\}, \{V_3, V_4, V_5\}\}, \qquad (33)\\
\mathcal{A}_0^+ = &\{\{V_1, V_2, V_3\}, \{V_1, V_2, V_5\}, \{V_1, V_3, V_5\},\\
&\{V_2, V_3, V_5\}, \{V_1, V_4\},\\
&\{V_2, V_4\}, \{V_3, V_4\}, \{V_4, V_5\}\}. \qquad (34)
\end{aligned}$$

Then, the cumulative map ψ_{Γ_2} is constructed as follows:

$$V_1 = \psi_{\Gamma_2}(1) = \left\{W_4^{(8)}, W_6^{(8)}, W_7^{(8)}, W_8^{(8)}\right\}, \tag{35}$$
$$V_2 = \psi_{\Gamma_2}(2) = \left\{W_3^{(8)}, W_5^{(8)}, W_7^{(8)}, W_8^{(8)}\right\}, \tag{36}$$
$$V_3 = \psi_{\Gamma_2}(3) = \left\{W_2^{(8)}, W_5^{(8)}, W_6^{(8)}, W_8^{(8)}\right\}, \tag{37}$$
$$V_4 = \psi_{\Gamma_2}(4) = \left\{W_1^{(8)}, W_2^{(8)}, W_3^{(8)}, W_4^{(8)}\right\}, \tag{38}$$
$$V_5 = \psi_{\Gamma_2}(5) = \left\{W_1^{(8)}, W_5^{(8)}, W_6^{(8)}, W_7^{(8)}\right\}, \tag{39}$$

which attains that $\tilde{\rho} = \rho^* = 4$. On the other hand, the modified cumulative map ψ'_{Γ_2} is given by

$$V_1 = \psi'_{\Gamma_2}(1) = \left\{W_1^{(7)}, W_9^{(7)}\right\}, \tag{40}$$
$$V_2 = \psi'_{\Gamma_2}(2) = \left\{W_2^{(7)}, W_8^{(7)}\right\}, \tag{41}$$
$$V_3 = \psi'_{\Gamma_2}(3) = \left\{W_3^{(7)}, W_7^{(7)}\right\}, \tag{42}$$
$$V_4 = \psi'_{\Gamma_2}(4) = \left\{W_4^{(7)}, W_6^{(7)}, W_7^{(7)}, W_8^{(7)}, W_9^{(7)}\right\}, \tag{43}$$
$$V_5 = \psi'_{\Gamma_2}(5) = \left\{W_5^{(7)}, W_6^{(7)}\right\}. \tag{44}$$

Observe that the rates of ψ'_{Γ_2} are given by $\tilde{\rho} = 13/5$, $\rho^* = 5$. Hence, ψ'_{Γ_2} gives smaller $\tilde{\rho}$ but larger ρ^* than ψ_{Γ_2}. □

As shown in Examples 6 and 7, the modified cumulative map cannot always overcome the defects of the original cumulative maps. Hence, in the next section, we propose a construction method of multiple assignment maps that can attain the optimal average or worst case coding rates based on integer programming.

3. Optimal Multiple Assignment Maps

For a multiple assignment map $\varphi_\Gamma : \{1, 2, \ldots, n\} \to 2^{\boldsymbol{W}_{(t,m)}}$, a set $\boldsymbol{A} \subseteq \boldsymbol{V}$, and $p \in \{0, 1, \ldots, 2^n - 1\}$, let $\boldsymbol{X}_p$ be the subset of $\boldsymbol{W}_{(t,m)}$ defined by

$$\boldsymbol{X}_p = \left[\bigcap_{i:b(p)_i=1} \varphi_\Gamma(i)\right] \cap \left[\bigcap_{i:b(p)_i=0} \overline{\varphi_\Gamma(i)}\right], \tag{45}$$

where $b(p)_i$ is the i-th least significant bit in the n-bit binary representation of p. For example, in the case of $p = 4$ and $n = 3$, it holds that $b(4)_3 = 1$, and (45) becomes $\boldsymbol{X}_4 = \varphi_\Gamma(3) \cap \overline{\varphi_\Gamma(1)} \cap \overline{\varphi_\Gamma(2)}$. Figure 1 is the Venn diagram which shows the relation between $\boldsymbol{X}_p$'s and $\varphi_\Gamma(i)$'s in the case of $n = 3$. Since φ_Γ must satisfy (13), it must hold that $\bigcap_{i=1}^n \overline{\varphi_\Gamma(i)} = \emptyset$, which implies that $\boldsymbol{X}_0 = \emptyset$. Hence, we consider only $\boldsymbol{X}_p$ for $p = 1, 2, \ldots, 2^n - 1$ in the following.

Then, it is easy to check that $\boldsymbol{X}_p$'s satisfy the following equations for an arbitrary n and $N \overset{\text{def}}{=} 2^n - 1$.

$$\boldsymbol{X}_p \cap \boldsymbol{X}_{p'} = \emptyset \quad \text{if } p \neq p' \tag{46}$$
$$\varphi_\Gamma(i) = \bigcup_{p:b(p)_i=1} \boldsymbol{X}_p \tag{47}$$
$$\Phi_\Gamma(\boldsymbol{A}) = \bigcup_{V_i \in \boldsymbol{A}} \varphi_\Gamma(i) = \bigcup_{\substack{p:b(p)_i=1 \\ \text{for some } V_i \in \boldsymbol{A}}} \boldsymbol{X}_p \tag{48}$$

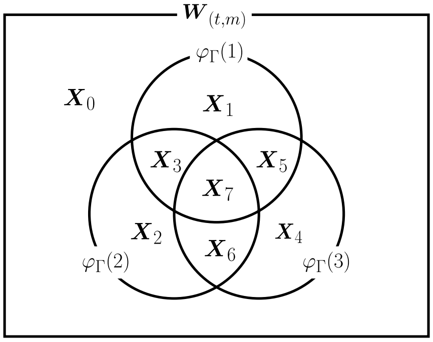

Fig. 1 Relation between $\varphi_\Gamma(i)$'s and $\boldsymbol{X}_k$'s in the case of $n = 3$.

Letting $x_p = |\boldsymbol{X}_p|$, the cardinality of $\Phi_\Gamma(\boldsymbol{A})$ is given by

$$|\Phi_\Gamma(\boldsymbol{A})| = \sum_{\substack{p:b(p)_i=1 \\ \text{for some } V_i \in \boldsymbol{A}}} x_p, \tag{49}$$

from (46) and (48).

Now, we describe how to design the optimal multiple assignment map $\tilde{\varphi}_\Gamma$ which attains the minimum average coding rate. Note that, in order to design the multiple assignment map φ_Γ for the set of primitive shares $\boldsymbol{W}_{(t,m)}$, we have to determine only x_p, $p = 1, 2, \ldots, N$, and t, since m can be calculated as $m = \sum_{p=1}^{N} x_p$ from (13) and (49).

Let $\boldsymbol{y} \overset{\text{def}}{=} [t, x_1, x_2, \ldots, x_N]$ be the $(N+1)$-dimensional parameter vector to minimize the average coding rate. Furthermore, for an integer ℓ and a share set $\boldsymbol{A}$, define an $(N+1)$-dimensional row vector $\boldsymbol{a}(\ell; \boldsymbol{A}) \overset{\text{def}}{=} [\ell, 1(\boldsymbol{A})_1, 1(\boldsymbol{A})_2, \ldots, 1(\boldsymbol{A})_N]$ where

$$1(\boldsymbol{A})_p = \begin{cases} 1 & \text{if } b(p)_i = 1 \text{ for some } V_i \in \boldsymbol{A} \\ 0 & \text{otherwise.} \end{cases} \tag{50}$$

Then, since (49) can be represented by inner product as $|\Phi_\Gamma(\boldsymbol{A})| = \boldsymbol{a}(0; \boldsymbol{A}) \cdot \boldsymbol{y}^\top$ where $\boldsymbol{y}^\top$ means the transpose of vector $\boldsymbol{y}$, the inequalities in the constraints (11) and (12) can be represented by $\boldsymbol{a}(0; \boldsymbol{A}) \cdot \boldsymbol{y}^\top \geq t$, and $\boldsymbol{a}(0; \boldsymbol{A}) \cdot \boldsymbol{y}^\top \leq t - 1$, respectively. Therefore, these constraints can be expressed as

$$\boldsymbol{a}(-1; \boldsymbol{A}) \cdot \boldsymbol{y}^\top \geq 0 \quad \text{if } \boldsymbol{A} \in \mathscr{A}_1^-, \tag{51}$$

$$-\boldsymbol{a}(-1; \boldsymbol{A}) \cdot \boldsymbol{y}^\top \geq 1 \quad \text{if } \boldsymbol{A} \in \mathscr{A}_0^+, \tag{52}$$

respectively. Furthermore, denoting the Hamming weight in the binary representation of p by h_p, it holds from (47) that

$$\sum_{i=1}^{n} |\varphi_\Gamma(i)| = \sum_{i=1}^{n} \sum_{p:b(p)_i=1} x_p = \sum_{p=1}^{N} h_p x_p = \boldsymbol{h} \cdot \boldsymbol{y}^\top, \tag{53}$$

where $\boldsymbol{h} = [0, h_1, \ldots, h_N] \in \mathbb{Z}^{N+1}$. Hence, the average coding rate $\tilde{\rho}$ in (14) is given by $(1/n)\ \boldsymbol{h} \cdot \boldsymbol{y}^\top$ which we want to minimize.

We note here that $\boldsymbol{a}(\cdot; \cdot)$ and $\boldsymbol{h}$ do not depend on the multiple assignment map φ_Γ, and hence, summarizing (50)–(53), we can formulate the integer programming problem $\text{IP}_{\tilde{\rho}}(\Gamma)$ that minimizes the average coding rate $\tilde{\rho}$ under the constraints of (11) and (12) as follows:

$$\begin{array}{llll} \underline{\text{IP}_{\tilde{\rho}}(\Gamma)} & & & \\ \text{minimize} & \boldsymbol{h} \cdot \boldsymbol{y}^\top & & \\ \text{subject to} & \boldsymbol{a}(-1; \boldsymbol{A}) \cdot \boldsymbol{y}^\top \geq 0 & \text{for} & \boldsymbol{A} \in \mathscr{A}_1^- \\ & -\boldsymbol{a}(-1; \boldsymbol{A}) \cdot \boldsymbol{y}^\top \geq 1 & \text{for} & \boldsymbol{A} \in \mathscr{A}_0^+ \\ & \boldsymbol{y} \geq \boldsymbol{0} & & \end{array}$$

The optimal multiple assignment map $\tilde{\varphi}_\Gamma$ that attains the minimum average coding rate can be constructed as follows. First, let $\tilde{\boldsymbol{y}} = [\tilde{t}, \tilde{x}_1, \tilde{x}_2, \ldots, \tilde{x}_N]$ be the minimizers of the integer programming problem $\text{IP}_{\tilde{\rho}}(\Gamma)$, and we use the $(\tilde{t}, \tilde{m})$-threshold SSS with primitive shares $\boldsymbol{W}_{(\tilde{t},\tilde{m})} = \{W_1^{(\tilde{t})}, W_2^{(\tilde{t})}, \ldots, W_{\tilde{m}}^{(\tilde{t})}\}$ for secret S where $\tilde{m}$ can be calculated from $\tilde{m} = \sum_{p=1}^{N} \tilde{x}_p$. Then, for each p, we can assign $\tilde{x}_p$ different primitive shares of $\boldsymbol{W}_{(\tilde{t},\tilde{m})}$ to $\boldsymbol{X}_p$ that satisfies $|\boldsymbol{X}_p| = \tilde{x}_p$ and (46). Finally, the multiple assignment map $\tilde{\varphi}_\Gamma$ is obtained by (47). Since $\tilde{\boldsymbol{y}}$ satisfies (51) and (52), it is easy to see that $V_i = \tilde{\varphi}_\Gamma(i)$ satisfies (11) and (12), and therefore, they satisfy (7) and (8). Summarizing, we can obtain the SSS with access structure Γ which has the optimal average coding rate in all the multiple assignment maps.

Next, we consider the integer programming problem $\text{IP}_{\rho^*}(\Gamma)$ that minimizes the worst coding rate ρ^*. Let M be the maximal number of assigned primitive shares among all V_i, $i = 1, 2, \ldots, n$. Then, it must hold that $|\varphi_\Gamma(i)| \leq M$ for all $i = 1, 2, \ldots, n$, and the minimization of M attains the optimal worst coding rate. Now, let $\boldsymbol{z}$ be the $(N+2)$-dimensional parameter vector defined by $\boldsymbol{z} \overset{\text{def}}{=} [M, t, x_1, x_2, \ldots, x_N]$. Then, it holds that $M = \boldsymbol{e} \cdot \boldsymbol{z}^\top$ where $\boldsymbol{e}$ is the $(N+2)$-dimensional row vector defined by $\boldsymbol{e} \overset{\text{def}}{=} [1, 0, 0, \ldots, 0]$. Furthermore, by defining $\boldsymbol{b}(\ell, \ell'; \boldsymbol{A}) \overset{\text{def}}{=} [\ell, \ell', 1(\boldsymbol{A})_1, 1(\boldsymbol{A})_2, \ldots, 1(\boldsymbol{A})_N]$ where $1(\boldsymbol{A})_p$ is defined by (50), the number of primitive shares assigned to a share set $\boldsymbol{A} \subseteq \boldsymbol{V}$ can be expressed as $\boldsymbol{b}(0, 0; \boldsymbol{A}) \cdot \boldsymbol{z}^\top$. This means that the condition $|\varphi_\Gamma(i)| \leq M$ can be represented by $-\boldsymbol{b}(-1, 0; \{V_i\}) \cdot \boldsymbol{z}^\top \geq 0$. Hence, in the same way as $\text{IP}_{\tilde{\rho}}(\Gamma)$, the integer programming problem $\text{IP}_{\rho^*}(\Gamma)$ that minimizes the worst coding rate ρ^* can be formulated as follows:

$$\begin{array}{llll} \underline{\text{IP}_{\rho^*}(\Gamma)} & & & \\ \text{minimize} & \boldsymbol{e} \cdot \boldsymbol{z}^\top & & \\ \text{subject to} & \boldsymbol{b}(0, -1; \boldsymbol{A}) \cdot \boldsymbol{z}^\top \geq 0 & \text{for} & \boldsymbol{A} \in \mathscr{A}_1^- \\ & -\boldsymbol{b}(0, -1; \boldsymbol{A}) \cdot \boldsymbol{z}^\top \geq 1 & \text{for} & \boldsymbol{A} \in \mathscr{A}_0^+ \\ & -\boldsymbol{b}(-1, 0; \{V\}) \cdot \boldsymbol{z}^\top \geq 0 & \text{for} & V \in \boldsymbol{V} \\ & \boldsymbol{z} \geq \boldsymbol{0} & & \end{array}$$

The multiple assignment map φ_Γ^* attaining the minimum ρ^* can also be constructed from the obtained minimizer in the same way as the construction of $\tilde{\varphi}_\Gamma$.

Remark 8: Actually, in SSSs, we can assume without loss of generality that $x_N = 0$, i.e., $\boldsymbol{X}_N = \bigcap_{i=1}^{n} \varphi_\Gamma(i) = \emptyset$ because it is not necessary to consider the set of primitive shares commonly contained in every share. Hence, the vectors in integer programming problems $\text{IP}_{\tilde{\rho}}(\Gamma)$ and $\text{IP}_{\rho^*}(\Gamma)$ can be reduced to N-dimensional and $(N+1)$-dimensional vectors, respectively. However, $x_N = 0$ does not hold generally in the case of ramp SSSs, which is described in Remark 22 in Sect. 5.2. □

Example 9: For the access structure Γ_1 defined by (18) and (19) in Example 3, the integer programming problem $\text{IP}_{\tilde{\rho}}(\Gamma_1)$ can be formulated as follows:

$$
\begin{aligned}
&\underline{\mathrm{IP}_{\tilde{\rho}}(\Gamma_1)}\\
&\text{minimize}\, x_1 + x_2 + 2x_3 + x_4 + 2x_5 + 2x_6 + 3x_7 + x_8\\
&\qquad +2x_9 + 2x_{10} + 3x_{11} + 2x_{12} + 3x_{13} + 3x_{14}\\
&\text{subject to}\quad -t + x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7\\
&\qquad\quad +x_9 + x_{10} + x_{11} + x_{12} + x_{13} + x_{14} \geq 0\\
&\qquad\quad -t + x_1 + x_3 + x_5 + x_7 + x_8 + x_9\\
&\qquad\qquad +x_{10} + x_{11} + x_{12} + x_{13} + x_{14} \geq 0\\
&\qquad\quad -t + x_2 + x_3 + x_6 + x_7 + x_8 + x_9\\
&\qquad\qquad +x_{10} + x_{11} + x_{12} + x_{13} + x_{14} \geq 0\\
&\qquad\quad -t + x_4 + x_5 + x_6 + x_7 + x_8 + x_9\\
&\qquad\qquad +x_{10} + x_{11} + x_{12} + x_{13} + x_{14} \geq 0\\
&\qquad\quad t - x_1 - x_2 - x_3 - x_5 - x_6 - x_7\\
&\qquad\qquad -x_9 - x_{10} - x_{11} - x_{13} - x_{14} \geq 1\\
&\qquad\quad t - x_1 - x_3 - x_4 - x_5 - x_6 - x_7\\
&\qquad\qquad -x_9 - x_{11} - x_{12} - x_{13} - x_{14} \geq 1\\
&\qquad\quad t - x_2 - x_3 - x_4 - x_5 - x_6 - x_7\\
&\qquad\qquad -x_{10} - x_{11} - x_{12} - x_{13} - x_{14} \geq 1\\
&\qquad\qquad t - x_8 - x_9 - x_{10} - x_{11}\\
&\qquad\qquad\qquad -x_{12} - x_{13} - x_{14} \geq 1\\
&\qquad\qquad x_p \geq 0,\ p = 1, 2, \ldots, 14
\end{aligned}
$$

By solving the above $\mathrm{IP}_{\tilde{\rho}}(\Gamma_1)$, we obtain that the value of the objective function is 5, which is attained by the following minimizers:

$$
\begin{aligned}
&\tilde{t} = 3,\\
&\tilde{x}_1 = \tilde{x}_2 = \tilde{x}_4 = 1,\ \tilde{x}_8 = 2,\\
&\tilde{x}_i = 0 \quad \text{for } i = 3, 5, 6, 7, 9, 10, \ldots, 14. \qquad (54)
\end{aligned}
$$

Hence, $\tilde{m}$ is given by $\tilde{m} = \sum_{p=1}^{14} \tilde{x}_p = 5$, and $\boldsymbol{X}_p$'s become $\boldsymbol{X}_1 = \{W_1^{(3)}\}$, $\boldsymbol{X}_2 = \{W_2^{(3)}\}$, $\boldsymbol{X}_4 = \{W_3^{(3)}\}$, and $\boldsymbol{X}_8 = \{W_4^{(3)}, W_5^{(3)}\}$ where $\boldsymbol{W}_{(3,5)} = \{W_1^{(3)}, W_2^{(3)}, \ldots, W_5^{(3)}\}$. Finally, from (47), $\tilde{\varphi}_{\Gamma_1}$ is constructed as

$$V_1 = \tilde{\varphi}_{\Gamma_1}(1) = \left\{W_1^{(3)}\right\}, \qquad (55)$$

$$V_2 = \tilde{\varphi}_{\Gamma_1}(2) = \left\{W_2^{(3)}\right\}, \qquad (56)$$

$$V_3 = \tilde{\varphi}_{\Gamma_1}(3) = \left\{W_3^{(3)}\right\}, \qquad (57)$$

$$V_4 = \tilde{\varphi}_{\Gamma_1}(4) = \left\{W_4^{(3)}, W_5^{(3)}\right\}. \qquad (58)$$

In this case, we have that $\tilde{\rho} = 5/4$ and $\rho^* = 2$. The integer programming problem $\mathrm{IP}_{\rho^*}(\Gamma_1)$ derives the same solutions as (54), and hence, it holds that $\tilde{\varphi}_{\Gamma_1} = \varphi^*_{\Gamma_1}$ in this example. Recall that the cumulative map ψ_{Γ_1} attains the coding rates $\tilde{\rho} = 9/4$ and $\rho^* = 3$, and the modified cumulative map ψ'_{Γ_1} attains $\tilde{\rho} = 5/2$ and $\rho^* = 4$. Hence, φ_{Γ_1} can attain smaller coding rates compared with ψ_{Γ_1} and ψ'_{Γ_1}. □

Example 10: For the access structure Γ_2 defined by (33) and (34) in Example 7, we can obtain the following multiple assignment map by solving the integer programming problem $\mathrm{IP}_{\tilde{\rho}}(\Gamma_2)$.

$$V_1 = \tilde{\varphi}_{\Gamma_2}(1) = \left\{W_1^{(4)}\right\}, \qquad (59)$$

$$V_2 = \tilde{\varphi}_{\Gamma_2}(2) = \left\{W_2^{(4)}\right\}, \qquad (60)$$

$$V_3 = \tilde{\varphi}_{\Gamma_2}(3) = \left\{W_3^{(4)}\right\}, \qquad (61)$$

$$V_4 = \tilde{\varphi}_{\Gamma_2}(4) = \left\{W_4^{(4)}, W_5^{(4)}\right\}, \qquad (62)$$

$$V_5 = \tilde{\varphi}_{\Gamma_2}(5) = \left\{W_6^{(4)}\right\}, \qquad (63)$$

where $W_i^{(4)} \in \boldsymbol{W}_{(4,6)}$. Then, it holds that $\tilde{\rho} = 6/5$ and $\rho^* = 2$. Furthermore, it holds that $\tilde{\varphi}_{\Gamma_2} = \varphi^*_{\Gamma_2}$ in this access structure. Recall again that the cumulative map ψ_{Γ_2} attains the coding rates $\tilde{\rho} = \rho^* = 4$, and the modified cumulative map ψ'_{Γ_2} attains $\tilde{\rho} = 13/5$ and $\rho^* = 5$. Hence, $\tilde{\varphi}_{\Gamma_2}$ is more efficient than ψ_{Γ_2} and ψ'_{Γ_2}. □

Since any access structure can be realized by the cumulative map (and the modified cumulative map), there exists at least one multiple assignment map for any access structure. Therefore, the next theorem holds obviously.

Theorem 11: For any access structure Γ that satisfies monotonicity (3) and (4), the integer programming problems $\mathrm{IP}_{\tilde{\rho}}(\Gamma)$ and $\mathrm{IP}_{\rho^*}(\Gamma)$ always have at least one feasible solution, and hence, there exists the optimal multiple assignment map. □

Remark 12: The integer programming problems are NP-hard, and hence, the proposed algorithms may take much time in solving for large $N = 2^n - 1$ where $n = |\boldsymbol{V}|$. However, several techniques to solve the integer programming efficiently have been developed [17] and their implemented softwares are easily available. For instance, in the case of $\mathrm{IP}_{\rho}(\Gamma_3)$ in Example 13 with $n = 6$, it can be solved within 0.1 seconds by a laptop computer with a software called `lp_solve`[†]. □

Example 13: Consider the following access structure Γ_3:

$$
\begin{aligned}
\mathcal{A}_1^- = \{&\{V_1, V_3, V_4, V_5\}, \{V_1, V_3, V_5, V_6\}, \{V_1, V_4, V_5, V_6\},\\
&\{V_3, V_4, V_5, V_6\}, \{V_1, V_2, V_3\}, \{V_1, V_2, V_5\},\\
&\{V_1, V_2, V_6\}, \{V_2, V_3, V_4\}, \{V_2, V_3, V_5\},\\
&\{V_2, V_3, V_6\}, \{V_2, V_4, V_5\},\\
&\{V_2, V_4, V_6\}, \{V_2, V_5, V_6\}\}, \qquad (64)\\
\mathcal{A}_0^+ = \{&\{V_1, V_3, V_4, V_6\}, \{V_1, V_2, V_4\}, \{V_1, V_3, V_5\},\\
&\{V_1, V_4, V_5\}, \{V_1, V_5, V_6\}, \{V_3, V_4, V_5\},\\
&\{V_3, V_5, V_6\}, \{V_4, V_5, V_6\},\\
&\{V_2, V_3\}, \{V_2, V_5\}, \{V_2, V_6\}\}. \qquad (65)
\end{aligned}
$$

Then, we obtain the following multiple assignment map by solving $\mathrm{IP}_{\tilde{\rho}}\,(\Gamma_3)$.

$$V_1 = \tilde{\varphi}_{\Gamma_3}(1) = \left\{W_1^{(6)}, W_2^{(6)}\right\}, \qquad (66)$$

$$V_2 = \tilde{\varphi}_{\Gamma_3}(2) = \left\{W_1^{(6)}, W_3^{(6)}, W_4^{(6)}, W_5^{(6)}\right\}, \qquad (67)$$

$$V_3 = \tilde{\varphi}_{\Gamma_3}(3) = \left\{W_6^{(6)}\right\}, \qquad (68)$$

$$V_4 = \tilde{\varphi}_{\Gamma_3}(4) = \left\{W_2^{(6)}, W_5^{(6)}\right\}, \qquad (69)$$

$$V_5 = \tilde{\varphi}_{\Gamma_3}(5) = \left\{W_3^{(6)}, W_7^{(6)}\right\}, \qquad (70)$$

[†] `lp_solve` is a well-known free (the GNU lesser general public license) linear programming software, which can also treat integer programming problems.

$$V_6 = \tilde{\varphi}_{\Gamma_3}(6) = \left\{W_8^{(6)}\right\}, \tag{71}$$

where $W_i^{(6)} \in \boldsymbol{W}_{(6,8)}$. $\tilde{\varphi}_{\Gamma_3}$ attains that $\tilde{\rho} = 2$ and $\rho^* = 4$. On the other hand, the cumulative map for the access structure Γ_3 are given by

$$\begin{aligned} V_1 &= \psi_{\Gamma_3}(1) \\ &= \left\{W_6^{(11)}, W_7^{(11)}, W_8^{(11)}, W_9^{(11)}, W_{10}^{(11)}, W_{11}^{(11)}\right\}, \end{aligned} \tag{72}$$

$$\begin{aligned} V_2 &= \psi_{\Gamma_3}(2) \\ &= \left\{W_1^{(11)}, W_3^{(11)}, W_4^{(11)}, W_5^{(11)}, \right. \\ &\qquad\qquad \left. W_6^{(11)}, W_7^{(11)}, W_8^{(11)}\right\}, \end{aligned} \tag{73}$$

$$\begin{aligned} V_3 &= \psi_{\Gamma_3}(3) \\ &= \left\{W_2^{(11)}, W_4^{(11)}, W_5^{(11)}, W_8^{(11)}, W_{10}^{(11)}, W_{11}^{(11)}\right\}, \end{aligned} \tag{74}$$

$$\begin{aligned} V_4 &= \psi_{\Gamma_3}(4) \\ &= \left\{W_3^{(11)}, W_5^{(11)}, W_7^{(11)}, W_9^{(11)}, W_{10}^{(11)}, W_{11}^{(11)}\right\}, \end{aligned} \tag{75}$$

$$V_5 = \psi_{\Gamma_3}(5) = \left\{W_1^{(11)}, W_2^{(11)}, W_9^{(11)}, W_{11}^{(11)}\right\}, \tag{76}$$

$$\begin{aligned} V_6 &= \psi_{\Gamma_3}(6) \\ &= \left\{W_2^{(11)}, W_3^{(11)}, W_4^{(11)}, W_6^{(11)}, W_9^{(11)}, W_{10}^{(11)}\right\}, \end{aligned} \tag{77}$$

where $W_i^{(11)} \in \boldsymbol{W}_{(11,11)}$. ψ_{Γ_3} has $\tilde{\rho} = 35/6$ and $\rho^* = 7$. Furthermore, the modified cumulative map for Γ_3 requires $(12, 15)$-threshold SSS and has $\tilde{\rho} = 5$ and $\rho^* = 9$. □

Next, we clarify what kind of access structure can be realized as an ideal SSS by the multiple assignment map.

Theorem 14: For an access structure Γ, the SSS constructed by the optimal multiple assignment map is ideal, i.e., $\rho_i = 1$ for all i, if and only if $\mathcal{A}_1^-$ of Γ can be represented by

$$\mathcal{A}_1^- = \bigcup_{\substack{\forall\{j_1, j_2, \ldots, j_t\} \\ \subseteq\{1,2,\ldots,m\}}} \left\{\boldsymbol{A}_{j_1} \times \boldsymbol{A}_{j_2} \times \cdots \times \boldsymbol{A}_{j_t}\right\}, \tag{78}$$

where t is a positive integer and $\{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_m\}$ is a partition of $\boldsymbol{V}$ which satisfies

$$\bigcup_{j=1}^{m} \boldsymbol{A}_j = \boldsymbol{V}, \tag{79}$$

$$\boldsymbol{A}_j \neq \emptyset \quad \text{for } j = 1, 2, \ldots, m, \tag{80}$$

$$\boldsymbol{A}_j \cap \boldsymbol{A}_{j'} = \emptyset \quad \text{if } j \neq j'. \tag{81}$$

□

Proof of Theorem 14: If there exists a partition $\{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_m\}$ satisfying (78)–(81) for the access structure Γ, the ideal SSS can be obtained by letting

$$\varphi_\Gamma(i) = W_j^{(t)} \in \boldsymbol{W}_{(t,m)} \quad \text{if } V_i \in \boldsymbol{A}_j \tag{82}$$

for each $i = 1, 2, \ldots, n$. Next, we show the necessity of (78)–(81). Suppose that a certain $\varphi_\Gamma(i)$ attains $\rho_i = 1$ for all i. Then, define each $\boldsymbol{A}_j$ as

$$\boldsymbol{A}_j \overset{\text{def}}{=} \Phi_\Gamma^{-1}\left(\left\{W_j^{(t)}\right\}\right), \; j = 1, 2, \ldots, m, \tag{83}$$

where $\Phi_\Gamma^{-1} : 2^{\boldsymbol{W}_{(t,m)}} \to 2^{\boldsymbol{V}}$ is the inverse map of $\Phi_\Gamma(\boldsymbol{A}) \overset{\text{def}}{=} \sum_{i:V_i \in \boldsymbol{A}} \varphi_\Gamma(i)$. Then, it is easy to see that $\boldsymbol{A}_j$'s satisfy (78), (79) and (80). Next, we prove that $\boldsymbol{A}_j$'s defined by (83) satisfy (81). Assume that there exist $\boldsymbol{A}_j$ and $\boldsymbol{A}_{j'}$, $j \neq j'$, not satisfying (81). Then, there exists a share $V_i \in \boldsymbol{A}_j \cap \boldsymbol{A}_{j'}$. This means that $W_j^{(t)}, W_{j'}^{(t)} \in \varphi_\Gamma(i)$, which contradicts $\rho_i = |\varphi_\Gamma(i)| = 1$. Hence, $\{\boldsymbol{A}_1, \boldsymbol{A}_2, \ldots, \boldsymbol{A}_m\}$ must be a partition of $\boldsymbol{V}$ satisfying (78)–(81). □

In the case of $t = 2$, it is known that an access structure Γ can be realized by an ideal SSS if and only if Γ can be represented by a complete multipartite graph [18]. We note that this condition coincides with (78)–(81) in this case. Furthermore, in the case that $|\boldsymbol{A}_j| = 1$ for $j = 1, 2, \ldots, m$, the access structure coincides with the (t, m)-threshold access structure. Hence, if Γ is the (k, n)-threshold access structure, the multiple assignment maps obtained from the integer programming problems $\mathrm{IP}_{\tilde{\rho}}(\Gamma)$ and $\mathrm{IP}_{\rho^*}(\Gamma)$ obviously satisfy that $|\tilde{\varphi}_\Gamma(i)| = |\varphi_\Gamma^*(i)| = 1$ for all i.

We note that any access structures not satisfying (78)–(81) must have $\tilde{\rho} > 1$ and $\rho^* \geq 2$ if the multiple assignment map is used. But, an access structure not satisfying (78)–(81) might be realized as an ideal SSS if we use another construction method. For example, refer [7].

In this paper, we assume that every share is significant. But, if there exist vacuous shares in the access structure Γ, it is cumbersome to check whether each share is significant or vacuous. From Remark 1, the optimal multiple assignment map $\tilde{\varphi}_\Gamma$ attaining the minimum average coding rate must satisfy that $|\tilde{\varphi}_\Gamma(i)| = 0$ for any vacuous share V_i. On the other hand, it clearly holds that $|\varphi_\Gamma(i)| \geq 1$ for every significant share V_i since $\rho_i \geq 1$ holds for any significant share. Hence, by solving the integer programming problem $\mathrm{IP}_{\tilde{\rho}}(\Gamma)$, we can also know whether a share is significant or vacuous although it cannot be known by solving $\mathrm{IP}_{\rho^*}(\Gamma)$.

4. Multiple Assignment Maps for Incomplete Access Structures

In the previous sections, we considered how to construct a SSS for a complete general access structure $\Gamma = \{\mathcal{A}_1, \mathcal{A}_0\}$. But in practice, it may be cumbersome to specify whether each subset of $\boldsymbol{V}$ is a qualified set or a forbidden set because the number of the subsets is 2^n. Hence, a method is proposed in [11] to construct a SSS for the case such that some subsets of $\boldsymbol{V}$ are not specified as qualified nor forbidden sets.

Theorem 15 ([11]): Let $\Gamma^\sharp = \{\mathcal{A}_1^\sharp, \mathcal{A}_0^\sharp\}$ be an incomplete access structure, which has $\mathcal{A}_1^\sharp \cup \mathcal{A}_1^\sharp \neq 2^{\boldsymbol{V}}$. Then, there exists a complete access structure $\Gamma = \{\mathcal{A}_1, \mathcal{A}_0\}$ such that

$$\mathcal{A}_1^\sharp \subseteq \mathcal{A}_1, \tag{84}$$

$$\mathcal{A}_0^\sharp \subseteq \mathcal{A}_0, \tag{85}$$

if and only if it holds that for any $\boldsymbol{A} \in \mathcal{A}_1^\sharp$ and $\boldsymbol{B} \in \mathcal{A}_0^\sharp$,

$$\boldsymbol{A} \nsubseteq \boldsymbol{B}. \tag{86}$$

□

In case that (86) is satisfied, the SSS satisfying the incomplete access structure $\Gamma^\sharp = \{\mathcal{A}_1^\sharp, \mathcal{A}_0^\sharp\}$ can be realized by applying the cumulative map to the complete access structure $\Gamma = \{\mathcal{A}_1, \mathcal{A}_0\}$. In fact, for the access structure $\Gamma^\sharp = \{\mathcal{A}_1^\sharp, \mathcal{A}_0^\sharp\}$, a SSS is constructed in [11] by a cumulative map $\psi_{\Gamma^\sharp}(i) = \bigcup_{j:V_i \notin \boldsymbol{F}_j}\{W_j^{(t)}\}$ for $\mathcal{A}_0^{\sharp+} = \{\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_m\}$. This construction corresponds to the case that

$$\mathcal{A}_0^+ = \mathcal{A}_0^{\sharp+} \text{ and } \mathcal{A}_1 = 2^{\boldsymbol{V}} - \mathcal{A}_0. \tag{87}$$

However, $\psi_{\Gamma^\sharp}$ is not efficient generally because $\psi_{\Gamma^\sharp}$ is a cumulative map, which is inefficient as described in Sect. 2.2. Furthermore, even if the cumulative map can attain the optimal coding rates for the access structure given by (87), the access structure may not be optimal among all the complete access structures $\Gamma = \{\mathcal{A}_1, \mathcal{A}_0\}$ satisfying (84) and (85) for the given $\Gamma^\sharp = \{\mathcal{A}_1^\sharp, \mathcal{A}_0^\sharp\}$.

In our construction based on integer programming, the optimal multiple assignment map for the incomplete access structure $\Gamma^\sharp = \{\mathcal{A}_1^\sharp, \mathcal{A}_0^\sharp\}$ can easily be obtained by applying $\mathrm{IP}_{\tilde{\rho}}(\Gamma)$ or $\mathrm{IP}_{\rho^*}(\Gamma)$ directly to $\Gamma^\sharp$.

Example 16: Let us consider the following access structure $\Gamma_3^\sharp = \{\mathcal{A}_1^\sharp, \mathcal{A}_0^\sharp\}$:

$$\begin{aligned}\mathcal{A}_1^\sharp &= \{\{V_1, V_4, V_5, V_6\}, \{V_1, V_2, V_5\}, \{V_1, V_2, V_6\},\\ &\qquad \{V_2, V_3, V_6\}, \{V_2, V_4, V_6\}\}, &(88)\\ \mathcal{A}_0^\sharp &= \{\{V_1, V_3, V_4, V_6\}, \{V_1, V_3, V_5\}, \{V_1, V_5, V_6\},\\ &\qquad \{V_3, V_4, V_5\}, \{V_4, V_5, V_6\}, \{V_2, V_5\}\}, &(89)\end{aligned}$$

Note that $\mathcal{A}_1^\sharp$ and $\mathcal{A}_0^\sharp$ satisfy $\mathcal{A}_1^\sharp \subseteq \mathcal{A}_1^-$ and $\mathcal{A}_0^\sharp \subseteq \mathcal{A}_0^+$ for $\Gamma_3 = \{\mathcal{A}_1, \mathcal{A}_0\}$, which is defined by (64) and (65) in Example 13. Then, by solving $\mathrm{IP}_{\tilde{\rho}}(\Gamma_3^\sharp)$, we obtain the following multiple assignment map.

$$V_1 = \tilde{\varphi}_{\Gamma_3^\sharp}(1) = \left\{W_1^{(4)}\right\}, \tag{90}$$

$$V_2 = \tilde{\varphi}_{\Gamma_3^\sharp}(2) = \left\{W_2^{(4)}, W_3^{(4)}\right\}, \tag{91}$$

$$V_3 = \tilde{\varphi}_{\Gamma_3^\sharp}(3) = \left\{W_4^{(4)}\right\}, \tag{92}$$

$$V_4 = \tilde{\varphi}_{\Gamma_3^\sharp}(4) = \left\{W_4^{(4)}\right\}, \tag{93}$$

$$V_5 = \tilde{\varphi}_{\Gamma_3^\sharp}(5) = \left\{W_5^{(4)}\right\}, \tag{94}$$

$$V_6 = \tilde{\varphi}_{\Gamma_3^\sharp}(6) = \left\{W_6^{(4)}\right\}, \tag{95}$$

where $W_i^{(4)} \in \boldsymbol{W}_{(4,6)}$, and it holds that $\tilde{\rho} = 7/6$ and $\rho^* = 2$. Hence, we can realize a more efficient SSS for the incomplete access structure $\Gamma_3^\sharp$ rather than the complete access structure Γ_3 because $\Gamma_3^\sharp$ has fewer constraints. If we apply the cumulative map to $\Gamma_3^\sharp$, $\psi_{\Gamma_3^\sharp}$ is constructed from the (6, 6)-threshold scheme, and it has $\tilde{\rho} = 3$ and $\rho^* = 5$. □

Similarly to complete SSSs, vacuous shares V_i in $\Gamma^\sharp = \{\mathcal{A}_1^\sharp, \mathcal{A}_0^\sharp\}$ can be detected by checking $|\varphi_{\Gamma^\sharp}(i)| = 0$ for the solution of the $\mathrm{IP}_{\tilde{\rho}}(\Gamma^\sharp)$.

5. Ramp SSSs with General Access Structures

The coding rate ρ_i must satisfy $\rho_i \geq 1$ for any significant share V_i if the access structure is *perfect*, i.e., every subset $\boldsymbol{A} \subseteq \boldsymbol{V}$ is classified into either qualified sets $\mathcal{A}_1$ or forbidden sets $\mathcal{A}_0$. Note that all access structures considered in previous sections are perfect even if they are incomplete. But, in the case of ramp (i.e., non-perfect) access structures such that some subsets of $\boldsymbol{V}$ are allowed to have intermediate properties between the qualified and forbidden sets, it is possible to decrease the coding rate ρ_i to less than 1. The SSSs having the ramp access structure are called *ramp schemes* [19], [20]. In this section, we treat the construction of ramp SSSs based on the multiple assignment maps. We consider only the minimum average coding rate in this section. But, for the minimum worst coding rate, integer programming can be formulated in a similar way.

5.1 Preliminaries for Ramp Schemes

First, let us review the definition of ramp SSSs. Suppose that $L + 1$ families $\mathcal{A}_j \subseteq 2^{\boldsymbol{V}}$, $j = 0, 1, \ldots, L$, satisfy the following.

$$H(S|\boldsymbol{A}) = \frac{L-j}{L}H(S), \quad \text{for any } \boldsymbol{A} \in \mathcal{A}_j \tag{96}$$

Equation (96) implies that the secret S leaks out from a set $\boldsymbol{A} \in \mathcal{A}_j$ with the amount of $(j/L)H(S)$. Especially, S can be reproduced completely from any $\boldsymbol{A} \in \mathcal{A}_L$, and any $\boldsymbol{A} \in \mathcal{A}_0$ leaks out no information of S. Note that, in the case of $L = 1$, the ramp SSS reduces to the perfect SSS treated in Sects. 2–4, and hence, the ramp SSS can be considered as an extension of the perfect SSS. We call $\Gamma^R = \{\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_L\}$ the access structure of the ramp SSS with $L + 1$ levels. Without loss of generality, we can assume that $\bigcup_{j=0}^{L} \mathcal{A}_j = 2^{\boldsymbol{V}}$ and $\mathcal{A}_j \cap \mathcal{A}_{j'} = \emptyset$ for $j \neq j'$, although incomplete access structures with $\bigcup_{j=0}^{L} \mathcal{A}_j \neq 2^{\boldsymbol{V}}$ can be treated in the same way as in Sect. 4.

For example, the access structure of (k, L, n)-threshold ramp SSS [19], [20] is defined as follows:

$$\mathcal{A}_0 = \{\boldsymbol{A} \in 2^{\boldsymbol{V}} : 0 \leq |\boldsymbol{A}| \leq k - L\}, \tag{97}$$

$$\mathcal{A}_j = \{\boldsymbol{A} \in 2^{\boldsymbol{V}} : |\boldsymbol{A}| = k - L + j\}, \text{for } 1 \leq j \leq L - 1, \tag{98}$$

$$\mathcal{A}_L = \{\boldsymbol{A} \in 2^{\boldsymbol{V}} : k \leq |\boldsymbol{A}| \leq n\}. \tag{99}$$

In ramp SSSs, a significant share can also be defined in the same way as the perfect SSSs shown in Sect. 2.1. A share $V_i \in \boldsymbol{V}$ is called *significant* if there exists a share set $\boldsymbol{A} \in 2^{\boldsymbol{V}}$ such that $\boldsymbol{A} \cup \{V_i\} \in \mathcal{A}_j$ and $\boldsymbol{A} \in \mathcal{A}_{j'}$ with $j > j'$. Then, a non-significant share $V_{i'}$ satisfies that $\boldsymbol{A} \cup \{V_{i'}\} \in \mathcal{A}_j$ for any share set $\boldsymbol{A} \in \mathcal{A}_j$, $j = 0, 1, \ldots, L$. Furthermore, if a non-significant share $V_{i'}$ satisfies $\{V_{i'}\} \in \mathcal{A}_0$, $V_{i'}$ plays no roll in the ramp SSS, and hence, we call $V_{i'}$ a *vacuous* share. However, there exists a ramp scheme such that $\mathcal{A}_0 = \emptyset$ and a non-significant share satisfy $\{V_{i'}\} \in \mathcal{A}_j$ for some $j \geq 1$.

This case implies that $H(V_{i'}) \geq H(S)/L$, and $H(V_{i'}|V) = 0$ for any $V \in \boldsymbol{V}$, i.e., a non-significant $V_{i'}$ is included in every share. Therefore, we call such a non-significant share $V_{i'}$ a *common* share.

Remark 17: It is known that for any access structure with $L+1$ levels, the coding rate ρ_i must satisfy $\rho_i \geq 1/L$ for any significant share V_i [21]. Especially, in the case of (k, L, n)-threshold SSSs, the optimal ramp SSS attaining $\rho_i = 1/L$ for all i can easily be constructed [19], [20]. Any common share V_i must also satisfy that $\rho_i \geq 1/L$. On the other hand, in the same way as Remark 1 for the perfect SSSs, each vacuous share V_i can be realized as $\rho_i = 0$ for any access structure. Furthermore, if there exists a vacuous share with $\rho_i > 0$, the average coding rate can be reduced by setting $\rho_i = 0$ without changing all the significant and the common shares. □

Letting $\check{\mathcal{A}}_j \overset{\text{def}}{=} \bigcup_{\ell=j}^{L} \mathcal{A}_\ell$ and $\hat{\mathcal{A}}_j \overset{\text{def}}{=} \bigcup_{\ell=0}^{j} \mathcal{A}_\ell$, for $j = 0, 1, \ldots, L$, the monotonicity in (3) and (4) are extended as follows:

$$\boldsymbol{A} \in \check{\mathcal{A}}_j \;\Rightarrow\; \boldsymbol{A}' \in \check{\mathcal{A}}_j \text{ for all } \boldsymbol{A}' \supseteq \boldsymbol{A} \tag{100}$$

$$\boldsymbol{A} \in \hat{\mathcal{A}}_j \;\Rightarrow\; \boldsymbol{A}' \in \hat{\mathcal{A}}_j \text{ for all } \boldsymbol{A}' \subseteq \boldsymbol{A} \tag{101}$$

Therefore, the minimal and maximal families of the access structure, $\Gamma^{R-} = \{\mathcal{A}_0^-, \mathcal{A}_1^-, \ldots, \mathcal{A}_L^-\}$ and $\Gamma^{R+} = \{\mathcal{A}_0^+, \mathcal{A}_1^+, \ldots, \mathcal{A}_L^+\}$, respectively, can be defined as

$$\mathcal{A}_j^- = \{\boldsymbol{A} \in \mathcal{A}_j : \boldsymbol{A} - \{V\} \notin \check{\mathcal{A}}_j \text{ for any } V \in \boldsymbol{A}\}, \tag{102}$$

$$\mathcal{A}_j^+ = \{\boldsymbol{A} \in \mathcal{A}_j : \boldsymbol{A} \cup \{V\} \notin \hat{\mathcal{A}}_j \text{ for any } V \in 2^{\boldsymbol{V}} - \boldsymbol{A}\}. \tag{103}$$

Then, the following theorem holds.

Theorem 18 ([21]): A ramp SSS with access structure $\Gamma^R = \{\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_L\}$ can be constructed if and only if $\check{\mathcal{A}}_j$ and $\hat{\mathcal{A}}_j$ satisfy the monotonicity (100) and (101), respectively, for all $j = 1, 2, \ldots, L$. □

In Theorem 18, the necessity of the condition is obvious, and the sufficiency is established by the next construction.

Construction 19 ([21]): Let $S = \{S^{\langle 1\rangle}, S^{\langle 2\rangle}, \ldots, S^{\langle L\rangle}\}$ be a secret, and let $\Gamma^{\langle j\rangle} = \{\check{\mathcal{A}}_j, 2^{\boldsymbol{V}} - \check{\mathcal{A}}_j\}$, $j = 1, 2, \ldots, L$, be the perfect access structures determined from a given access structure Γ^R. Since each $\Gamma^{\langle j\rangle}$ is a perfect access structure satisfying the monotonicity (3) and (4), we can construct a SSS with $\Gamma^{\langle j\rangle}$ for secret $S^{\langle j\rangle}$. Letting $\{V_1^{\langle j\rangle}, V_2^{\langle j\rangle}, \ldots, V_n^{\langle j\rangle}\}$ be the shares for $S^{\langle j\rangle}$ and $\Gamma^{\langle j\rangle}$, the share $V_i = \{V_i^{\langle 1\rangle}, V_i^{\langle 2\rangle}, \ldots, V_i^{\langle L\rangle}\}$ realizes the access structure Γ^R. For Γ^R, a ramp SSS can also be constructed from $\{2^{\boldsymbol{V}} - \hat{\mathcal{A}}_j, \hat{\mathcal{A}}_j\}$ instead of $\Gamma^{\langle j\rangle} = \{\check{\mathcal{A}}_j, 2^{\boldsymbol{V}} - \check{\mathcal{A}}_j\}$. □

Remark 20: Note that in Construction 19, we have $\rho_i \geq 1$ for any access structure. For example, in the case that Construction 19 is applied to the (k, L, n)-threshold access structure, the constructed ramp SSS has $\rho_i = 1$ although the (k, L, n)-threshold SSS can be realized with $\rho_i = 1/L$. Therefore, Construction 19 is not efficient generally. □

Example 21: Consider the following ramp access structure Γ_4^R for $\boldsymbol{V} = \{V_1, V_2, V_3, V_4\}$:

$$\mathcal{A}_3 = \{\{V_1, V_2, V_3, V_4\}\}, \tag{104}$$

$$\mathcal{A}_2 = \{\{V_1, V_2, V_3\}, \{V_1, V_3, V_4\}\}, \tag{105}$$

$$\mathcal{A}_1 = \{\{V_1, V_2, V_4\}, \{V_2, V_3, V_4\}\}, \tag{106}$$

$$\mathcal{A}_0 = \{\boldsymbol{A} : 0 \leq |\boldsymbol{A}| \leq 2\}. \tag{107}$$

First, we derive the access structures $\Gamma^{\langle 1\rangle}$, $\Gamma^{\langle 2\rangle}$, and $\Gamma^{\langle 3\rangle}$ based on (104)–(107), and it is easy to see that $\Gamma^{\langle 1\rangle}$ and $\Gamma^{\langle 3\rangle}$ become (3, 4)- and (4, 4)-threshold access structures, respectively. Hence, we have $V_i^{\langle 1\rangle} = W_i^{(3)}$ and $V_i^{\langle 3\rangle} = W_i^{(4)}$ for $i = 1, 2, 3, 4$ where $\{W_i^{(3)}\}_{i=1}^4$ and $\{W_i^{(4)}\}_{i=1}^4$ are the share sets of (3, 4)- and (4, 4)-threshold access structures for secrets $S^{\langle 1\rangle}$ and $S^{\langle 3\rangle}$, respectively. Furthermore, a perfect SSS with the access structure $\Gamma^{\langle 2\rangle}$ for a secret $S^{\langle 2\rangle}$ can be realized by $\{V_i^{\langle 2\rangle}\}_{i=1}^4$ such that $V_1^{\langle 2\rangle} = W_1'^{(3)}$, $V_2^{\langle 2\rangle} = W_2'^{(3)}$, $V_3^{\langle 2\rangle} = W_3'^{(3)}$, and $V_4^{\langle 2\rangle} = W_2'^{(3)}$ where $\{W_i'^{(3)}\}_{i=1}^3$ is the share sets of (3, 3)-threshold SSS for $S^{\langle 2\rangle}$.

According to Construction 19, we can obtain the shares such that $V_1 = \{W_1^{(3)}, W_1'^{(3)}, W_1^{(4)}\}$, $V_2 = \{W_2^{(3)}, W_2'^{(3)}, W_2^{(4)}\}$, $V_3 = \{W_3^{(3)}, W_3'^{(3)}, W_3^{(4)}\}$, and $V_4 = \{W_4^{(3)}, W_4'^{(3)}, W_4^{(4)}\}$. Since each share consists of three primitive shares for three secrets $S^{\langle 1\rangle}$, $S^{\langle 2\rangle}$, $S^{\langle 3\rangle}$, the constructed ramp SSS has $\tilde{\rho} = \rho^* = 1$. □

The construction of ramp SSSs for general access structures are treated in [22]. But, since the construction in [22] is based on monotone span programming, it is much complicated compared with the multiple assignment map.

5.2 Optimal Multiple Assignment Maps for Ramp SSSs

First, let $\boldsymbol{W}_{(t,L,m)} = \{W_1^{(t,L)}, W_2^{(t,L)}, \ldots, W_m^{(t,L)}\}$ be the entire set of primitive shares for the (t, L, m)-threshold ramp SSS with the coding rate $\rho_i = 1/L$. Then, defining $\boldsymbol{y}$ and $\boldsymbol{a}(\ell; \boldsymbol{A})$ in the same way as the perfect SSSs in Sect. 3, the optimal ramp SSS by the multiple assignment map for a general ramp access structure Γ^R can be obtained by solving the following integer programming problem:

$$\begin{array}{rl}
\underline{\mathrm{IP}_{\tilde{\rho}}^{R}\left(\Gamma^{R}\right)} & \\
\text{minimize} & \boldsymbol{h} \cdot \boldsymbol{y}^{\top} \\
\text{subject to} & \boldsymbol{a}(-1; \boldsymbol{A}) \cdot \boldsymbol{y}^{\top} \geq 0 \text{ for } \boldsymbol{A} \in \mathcal{A}_L^- \\
 & -\boldsymbol{a}(-1; \boldsymbol{A}) \cdot \boldsymbol{y}^{\top} = j \text{ for } \boldsymbol{A} \in \mathcal{A}_j^+ \cup \mathcal{A}_j^-, \\
 & \qquad 1 \leq j \leq L-1 \ (\star) \\
 & -\boldsymbol{a}(-1; \boldsymbol{A}) \cdot \boldsymbol{y}^{\top} \geq L \text{ for } \boldsymbol{A} \in \mathcal{A}_0^+ \\
 & \boldsymbol{y} \geq \boldsymbol{0}
\end{array}$$

Remark 22: From the monotonicity defined in (100) and (101), it is sufficient to consider only $\boldsymbol{A} \in \mathcal{A}_j^+ \cup \mathcal{A}_j^-$ instead of all $\boldsymbol{A} \in \mathcal{A}_j$ on the marked line ($\star$) in $\mathrm{IP}_{\tilde{\rho}}^{R}\left(\Gamma^{R}\right)$. Note that the same primitive shares may be distributed to all shares since there may exist common shares in ramp SSSs. Hence, we may have $x_N \neq 0$ in the ramp SSSs although we can always assume that $x_N = 0$ in the perfect SSSs. □

From Remark 17, significant or common shares V_i

must satisfy that $|\varphi_\Gamma(i)| \geq 1$ for any multiple assignment map φ_Γ. On the other hand, $|\tilde{\varphi}_\Gamma(i')| = 0$ must hold for vacuous shares $V_{i'}$ for the optimal multiple assignment map $\tilde{\varphi}_\Gamma$ attaining the minimal average coding rate. Hence, it suffices to consider only significant shares and common shares in the ramp SSSs.

Example 23: If the access structures Γ_4^R in Example 21 is applied to the integer programming problem $\mathrm{IP}_{\tilde{\rho}}^{R}\left(\Gamma_4^R\right)$, the following multiple assignment map is obtained

$$V_1 = \tilde{\varphi}_{\Gamma_4^R}(1) = \left\{W_1^{(7,3)}, W_2^{(7,3)}\right\}, \tag{108}$$

$$V_2 = \tilde{\varphi}_{\Gamma_3^R}(2) = \left\{W_3^{(7,3)}, W_4^{(7,3)}\right\}, \tag{109}$$

$$V_3 = \tilde{\varphi}_{\Gamma_4^R}(3) = \left\{W_5^{(7,3)}, W_6^{(7,3)}\right\}, \tag{110}$$

$$V_4 = \tilde{\varphi}_{\Gamma_4^R}(4) = \left\{W_3^{(7,3)}, W_7^{(7,3)}\right\}, \tag{111}$$

where $W_i^{(7,3)} \in \boldsymbol{W}_{(7,3,7)}$. $\tilde{\varphi}_{\Gamma_4^R}$ attains that $\tilde{\rho} = \rho^* = 2/3$. □

Note that the coding rates less than 1 cannot be achieved by Construction 19. Furthermore, our construction is much simpler compared with the method in [22]. But, unfortunately, the integer programming problem may not have any feasible solutions in the case of ramp SSSs.

Example 24: The following access structure Γ_5^R cannot be constructed by any multiple assignment map since the corresponding integer programming problem has no feasible solution.

$$\begin{aligned}\mathcal{A}_4^- = \{&\{V_1, V_2, V_3, V_4\}, \{V_1, V_2, V_4, V_5\},\\ &\{V_2, V_3, V_4, V_5\}\},\end{aligned} \tag{112}$$

$$\begin{aligned}\mathcal{A}_3 = \{&\{V_1, V_2, V_3, V_5\}, \{V_1, V_3, V_4, V_5\}, \{V_1, V_2, V_3\},\\ &\{V_1, V_2, V_4\}, \{V_1, V_3, V_4\}, \{V_1, V_3, V_5\},\\ &\{V_2, V_3, V_4\}\},\end{aligned} \tag{113}$$

$$\begin{aligned}\mathcal{A}_2 = \{&\{V_1, V_2, V_5\}, \{V_1, V_4, V_5\}, \{V_2, V_3, V_5\},\\ &\{V_2, V_4, V_5\}, \{V_3, V_4, V_5\}, \{V_1, V_3\}, \{V_1, V_5\}\},\end{aligned} \tag{114}$$

$$\mathcal{A}_1 = \{\{V_1, V_2\}, \{V_2, V_3\}, \{V_3, V_4\}\}, \tag{115}$$

$$\mathcal{A}_0^+ = \{\{V_1, V_4\}, \{V_2, V_5\}, \{V_3, V_5\}\}, \tag{116}$$

□

In this case, we can modify the definition of the ramp SSS given by (96) as follows.

$$H(S|\boldsymbol{A}) = 0, \qquad \text{for all } \boldsymbol{A} \in \mathcal{A}_L, \tag{117}$$

$$H(S|\boldsymbol{A}) \geq \frac{L-j}{L} H(S), \ \text{for all } \boldsymbol{A} \in \mathcal{A}_j, 1 \leq j \leq L-1, \tag{118}$$

$$H(S|\boldsymbol{A}) = H(S), \qquad \text{for all } \boldsymbol{A} \in \mathcal{A}_0. \tag{119}$$

In order to implement (117)–(119) in the integer programming, it suffices to replace the marked line ($\star$) in $\mathrm{IP}_{\tilde{\rho}}^{R}\left(\Gamma^R\right)$ by $-\boldsymbol{a}(-1; \boldsymbol{A}_j) \cdot \boldsymbol{y}^\top \geq j$. Letting $\mathrm{IP}_{\tilde{\rho}}^{R2}\left(\Gamma^R\right)$ be the modified integer programming problem, the next theorem holds.

Theorem 25: The integer programming problem $\mathrm{IP}_{\tilde{\rho}}^{R2}\left(\Gamma^R\right)$ always has a feasible solution for any access structure Γ^R if Γ^R satisfies the monotonicity (100) and (101). □

Proof of Theorem 25: Let $\mathcal{V}$ be a multiset in $2^{\boldsymbol{V}}$, some elements of which may be the same. Then, for $\mathcal{V}$ and $\boldsymbol{A} \subseteq \boldsymbol{V}$, we define $N(\mathcal{V}, \boldsymbol{A})$ as follows.

$$N(\mathcal{V}, \boldsymbol{A}) = \left|\{\boldsymbol{A}' \in \mathcal{V} : \boldsymbol{A} \subseteq \boldsymbol{A}'\}\right|, \tag{120}$$

where all $\boldsymbol{A}' \in \mathcal{V}$ are treated as different sets even if some of them are the same. Now we construct a multiset $\mathcal{U}$ for $\Gamma^R = \{\mathcal{A}_0, \mathcal{A}_1, \ldots, \mathcal{A}_L\}$ by the next construction.

Construction 26:

(1) Let $\mathcal{U} := \emptyset$ and $j := 1$.
(2) For each $\boldsymbol{A} \in \mathcal{A}_{L-j}^+$ satisfying $N(\mathcal{U}, \boldsymbol{A}) < j$, we add $\boldsymbol{A}$ into $\mathcal{U}$, $(j - N(\mathcal{U}, \boldsymbol{A}))$ times.
(3) Let $j := j + 1$.
(4) If $j \leq L$, go to (2). In case of $j = L + 1$, go to (5).
(5) Output $\mathcal{U}$. □

From the monotonicity of $\check{\mathcal{A}}_j$ in (100), the family $\mathcal{U}$ can always be constructed. Then, letting $\mathcal{U} = \{\boldsymbol{F}_1, \boldsymbol{F}_2, \ldots, \boldsymbol{F}_m\}$, we can define a map $\check{\psi} : \{1, 2, \ldots, n\} \to 2^{\boldsymbol{W}_{(m,L,m)}}$ by

$$\check{\psi}(i) = \bigcup_{j:V_i \notin \boldsymbol{F}_j} \left\{W_j^{(m,L)}\right\}, \tag{121}$$

where $W_j^{(m,L)} \in \boldsymbol{W}_{(m,L,m)}$. Note that in the case of $L = 1$, (121) coincides with the cumulative map in (17). Furthermore, for any set $\boldsymbol{F}_\ell \in \mathcal{U}$, we can check from (121) that

$$W_{\ell'}^{(m,L)} \notin \bigcup_{i:V_i \in \boldsymbol{F}_\ell} \check{\psi}(i), \tag{122}$$

holds for all ℓ' satisfying $\boldsymbol{F}_\ell \subseteq \boldsymbol{F}_{\ell'}$.

Now, assume that $|\bigcup_{i:V_i \in \boldsymbol{A}} \check{\psi}(i)| \leq m - 1$ for $\boldsymbol{A} \in \mathcal{A}_L$. This implies that there exist at least one $W_\ell^{(m,L)} \notin \bigcup_{i:V_i \in \boldsymbol{A}} \check{\psi}(i)$. Hence, we have $\boldsymbol{A} \subseteq \boldsymbol{F}_\ell \in \mathcal{U}$ in the same way as we obtain (122), which contradicts the construction of $\mathcal{U}$ and the fact that $\boldsymbol{A} \in \mathcal{A}_L$. Therefore, we have $|\bigcup_{i:V_i \in \boldsymbol{A}} \check{\psi}(i)| = m$, i.e., $V_i = \check{\psi}(i)$ satisfies (117).

Next, assume that $\boldsymbol{F}_\ell \in \mathcal{A}_{L-j}^+$ for $j = 1, 2, \ldots, L$. Then, from Construction 26, there exists a family of j subsets $\{\boldsymbol{F}_{\ell_1}, \boldsymbol{F}_{\ell_2}, \ldots, \boldsymbol{F}_{\ell_j}\} \subseteq \mathcal{U}$ satisfying $\boldsymbol{F}_\ell \subseteq \boldsymbol{F}_{\ell'}$ for $\ell' \in \{\ell_1, \ell_2, \ldots, \ell_j\}$. Hence, it holds from (122) that $W_{\ell'}^{(m,L)} \notin \bigcup_{i:V_i \in \boldsymbol{F}_\ell} \check{\psi}(i)$ for $\ell' \in \{\ell_1, \ell_2, \ldots, \ell_j\}$. This means that $\left|\bigcup_{i:V_i \in \boldsymbol{F}_\ell} \check{\psi}(i)\right| \leq m - j$, and $V_i = \check{\psi}(i)$ satisfies (118) and (119).

Therefore, $\mathrm{IP}_{\tilde{\rho}}^{R2}\left(\Gamma^R\right)$ always has at least one feasible solution. □

Note that as shown in the following example, Construction 26 gives inefficient assignments of the primitive shares, generally.

Example 27: Assume that the access structure Γ_5^R in (112)–(116) satisfies the conditions (117)–(119). First, we apply Construction 26 to the access structure Γ_5^R. Then, we obtain the following multiset $\mathcal{U}_{\Gamma_5^R}$.

$$\begin{aligned}\mathcal{U}_{\Gamma_5^R} = \{&\{V_1, V_2, V_3, V_5\}, \{V_1, V_3, V_4, V_5\}, \{V_1, V_2, V_4\},\\ &\{V_1, V_2, V_5\}, \{V_1, V_4, V_5\}, \{V_2, V_3, V_5\},\\ &\{V_2, V_3, V_4\}, \{V_2, V_4, V_5\}, \{V_2, V_4, V_5\},\\ &\{V_3, V_4, V_5\}, \{V_1, V_4\}\}. \qquad (123)\end{aligned}$$

Hence, we can obtain $V_i = \check{\psi}(i)$, $i = 1, 2, \ldots, 5$, as follows:

$$\begin{aligned}V_1 &= \check{\psi}(1)\\ &= \left\{W_6^{(11,4)}, W_7^{(11,4)}, W_8^{(11,4)}, W_9^{(11,4)}, W_{10}^{(11,4)}\right\}, \qquad (124)\\ V_2 &= \check{\psi}(2) = \left\{W_2^{(11,4)}, W_5^{(11,4)}, W_{10}^{(11,4)}, W_{11}^{(11,4)}\right\}, \qquad (125)\\ V_3 &= \check{\psi}(3)\\ &= \left\{W_3^{(11,4)}, W_4^{(11,4)}, W_5^{(11,4)}, W_8^{(11,4)},\right.\\ &\qquad\qquad \left.W_9^{(11,4)}, W_{11}^{(11,4)}\right\}, \qquad (126)\\ V_4 &= \check{\psi}(4) = \left\{W_1^{(11,4)}, W_4^{(11,4)}, W_6^{(11,4)}\right\}, \qquad (127)\\ V_5 &= \check{\psi}(5) = \left\{W_3^{(11,4)}, W_7^{(11,4)}, W_{11}^{(11,4)}\right\}, \qquad (128)\end{aligned}$$

where $W_i \in \boldsymbol{W}_{(11,4,11)}$. In this case, we have $\tilde{\rho} = 21/20$ and $\rho^* = 3/2$ since it holds that $H(W_i^{(11,4)}) = H(S)/4$ for each i.

On the other hand, we can construct the following optimal multiple assignment map $\tilde{\varphi}_{\Gamma_5^R}$ by solving the integer programming problem $\mathrm{IP}_{\tilde{\rho}}^{R2}(\Gamma_5^R)$.

$$\begin{aligned}V_1 &= \tilde{\varphi}_{\Gamma_5^R}(1) = \left\{W_1^{(8,4)}, W_2^{(8,4)}\right\}, \qquad (129)\\ V_2 &= \tilde{\varphi}_{\Gamma_5^R}(2) = \left\{W_3^{(8,4)}, W_4^{(8,4)}, W_5^{(8,4)}\right\}, \qquad (130)\\ V_3 &= \tilde{\varphi}_{\Gamma_5^R}(3) = \left\{W_2^{(8,4)}, W_6^{(8,4)}\right\}, \qquad (131)\\ V_4 &= \tilde{\varphi}_{\Gamma_5^R}(4) = \left\{W_7^{(8,4)}, W_8^{(8,4)}\right\}, \qquad (132)\\ V_5 &= \tilde{\varphi}_{\Gamma_5^R}(5) = \left\{W_9^{(8,4)}\right\}, \qquad (133)\end{aligned}$$

where $W_i^{(8,4)} \in \boldsymbol{W}_{(8,4,9)}$, and it holds that $\tilde{\rho} = 1/2$ and $\rho^* = 3/4$, which are more efficient than the rates of Construction 26. Note that (124)–(128) and (129)–(133) do not satisfy (96) but satisfy (117)–(119). For instance, in (129)–(133), it holds for $\{V_1, V_5\} \in \mathcal{A}_2$ that $H(S|\{V_1, V_5\}) = H(S) > H(S)/2$.

Finally, we compare Construction 19 with Construction 26 for the access structure Γ_5^R. If we use the cumulative map to realize each perfect SSS with the access structure $\Gamma_5^{\langle j \rangle}$, $j = 1, 2, 3, 4$, in Construction 19, we obtain $\tilde{\rho} = 9/5$ and $\rho^* = 2$. Hence, Construction 19 is less efficient than Construction 26 in this case. □

6. Conclusion

We proposed a method to construct a SSS for any given general access structure based on (t, m)-threshold SSS and integer programming. The proposed method can attain the *optimal* average and/or worst coding rates in the sense of multiple assignment maps. Hence, the proposed method can attain smaller coding rates compared with the cumulative maps and the modified cumulative maps. Furthermore, the proposed method can be applied to incomplete and/or ramp access structures in addition to complete and perfect access structures.

Acknowledgments

The work of M. Iwamoto was partially supported by the MEXT Grant-in-Aid for Young Scientists (B) No.17760298, and the work of H. Yamamoto was partially supported by the MEXT Grant-in-Aid for Scientific Research on Priority Areas No.16016224.

References

[1] A. Shamir, "How to share a secret," Commun. ACM, vol.22, no.11, pp.612–613, 1979.

[2] G.R. Blakley, "Safeguarding cryptographic keys," AFIPS 1979 Nat. Computer Conf., vol.48, pp.313–317, 1979.

[3] E.D. Karnin, J.W. Greene, and M.E. Hellman, "On secret sharing systems," IEEE Trans. Inf. Theory, no.29, pp.35–41, 1983.

[4] R.M. Capocelli, A.D. Santis, L. Gargano, and U. Vaccaro, "On the size of shares for secret sharing schemes," J. Cryptol., vol.6, pp.157–167, 1993.

[5] L. Csirmaz, "The size of a share must be large," J. Cryptol., vol.10, pp.223–231, 1997.

[6] J. Benaloh and J. Leichter, "Generalized secret sharing and monotone functions," Advances in Cryptology-CRYPTO'88, LNCS 403, pp.27–35, Springer-Verlag, 1990.

[7] D.R. Stinson, "Decomposition construction for secret-sharing schemes," IEEE Trans. Inf. Theory, vol.40, no.1, pp.118–125, 1994.

[8] K. Tochikubo, T. Uyematsu, and R. Matusmoto, "Efficient secret sharing schemes based on authorized subsets," IEICE Trans. Fundamentals, vol.E88–A, no.1, pp.322–326, Jan. 2005.

[9] M. Itoh, A. Saito, and T. Nishizeki, "Secret sharing scheme realizing general access structure," IEEE Globecom, pp.99–102, 1987.

[10] M. Itoh, A. Saito, and T. Nishizeki, "Secret sharing scheme realizing general access structure," IEICE Trans. Fundamentals (Japanese Edition), vol.J71–A, no.8, pp.1592–1598, Aug. 1988.

[11] M. Itoh, A. Saito, and T. Nishizeki, "Multiple assignment scheme for sharing secret," J. Cryptol., vol.6, pp.15–20, 1993.

[12] G. Ateniese, C. Blundo, A.D. Santis, and D.R. Stinson, "Visual cryptography for general access structures," Inf. Comput., vol.129, pp.86–106, 1996.

[13] H. Koga, M. Iwamoto, and H. Yamamoto, "An analytic construction of the visual secret sharing scheme for color images," IEICE Trans. Fundamentals, vol.E84–A, no.1, pp.262–272, Jan. 2001.

[14] K. Tochikubo, "Efficient secret sharing schemes realizing general access structures," IEICE Trans. Fundamentals, vol.E87–A, no.7, pp.1788–1797, Jan. 2004.

[15] K. Koyama, "Cryptographic key sharing methods for multi-groups and security analysis," IECE Trans., vol.E66, no.1, pp.123–20, Jan. 1983.

[16] G.J. Simmons, W.A. Jackson, and K. Martin, "The geometry of shared secret schemes," Bulletin of the ICA, vol.1, no.2, pp.230–236, 1991.

[17] L.A. Wolsey, Integer Programming, Wiley Publishers, 1998.

[18] C. Blundo, A.D. Santis, D.R. Stinson, and U. Vaccaro, "Graph decompositions and secret sharing schemes," J. Cryptol., vol.8, pp.39–64, 1995.

[19] G.R. Blakley and C. Meadows, "Security of ramp schemes," Advances in Cryptology-CRYPTO'84, LNCS 196, pp.242–269, Springer-Verlag, 1985.

[20] H. Yamamoto, "On secret sharing systems using (k, L, n) threshold scheme," IEICE Trans. Fundamentals (Japanese Edition), vol.J68–A, no.9, pp.945–952, Sept. 1985. English translation: Electron. Commun. Jpn. 1, Commun., vol.69, no.9, pp.46–54, Scripta Technica, Inc., 1986.

[21] K. Kurosawa, K. Okada, K. Sakano, W. Ogata, and T. Tsujii, "Nonperfect secret sharing schemes and matroids," Advances in Cryptology-EUROCRYPT'93, LNCS 765, pp.126–141, Springer-Verlag, 1993.

[22] K. Srinathan, N.T. Rajan, and C.P. Rangan, "Non-perfect secret sharing over general access structures," Progress in Cryptology-INDOCRYPT'02, LNCS 2551, pp.409–421, Springer-Verlag, 2002.

Mitsugu Iwamoto was born in Fukuoka, on 29 July, 1976. He received the B.E., M.E., and Ph.D. degrees from the University of Tokyo, Japan, in 1999, 2001, and 2004, respectively. Currently, he is a research associate in Graduate School of Information Systems, University of Electro-Communications. His research interests include information security and cryptography. Dr. Iwamoto is a member of the IEEE and the SITA (Society of Information Theory and Its Applications).

Hirosuke Yamamoto was born in Wakayama, Japan, on November 15, 1952, He received the B.E. degree from Shizuoka University, in 1975 and the M.E. and Ph.D. degrees from the University of Tokyo, in 1977 and 1980, respectively, all in electrical engineering. In 1980, he joined Tokushima University. He was an Associate Professor at Tokushima University, the University of Electro- Communications, and the University of Tokyo, from 1983 to 1987, from 1987 to 1993, and from 1993 to 1999, respectively. Since 1999, he has been a Professor at the University of Tokyo. He was with the School of Engineering and the School of Information Science and Technology from 1993 to 1999 and from 1999 to 2004, respectively, and is now with the School of Frontier Sciences in the University of Tokyo. In 1989 and 1990, he was a Visiting Scholar at the Information Systems Laboratory, Stanford University. His research interests are in Shannon theory, data compression algorithms, and cryptology. Prof. Yamamoto is a member of the IEEE and the SITA (Society of Information Theory and its Applications). He served as Chair of the IEEE Information Theory Society Japan Chapter in 2002 and 2003, Vice President of the SITA in 2004 and 2005, and Technical Program Committee Co-Chair for the 2004 International Symposium on Information Theory and its Applications (ISITA2004), which was held at Parma, Italy.

Hirohisa Ogawa was born in Japan on 1971. His research interests include Cryptography and Information Security. He is currently with C4 Technology, Inc.