Electronics and Communications in Japan, Part 1, Vol. 69, No. 9, 1986
Translated from Tsushin Denshi Gakkai Ronbunshi, Vol. 68-A, No. 9, September 1985, pp. 945-952

Secret Sharing System Using (k, L, n) Threshold Scheme

Hirosuke Yamamoto, Member

Faculty of Engineering, Tokushima University, Tokushima, Japan 770

SUMMARY

In the (k, n) threshold scheme, the information X is partitioned and coded into subinformation. If any k subinformation is obtained among n subinformation, the original information X can be recovered completely. However, no information can be obtained at all concerning X from any $(k - 1)$ subinformation. Thus, the (k, n) threshold scheme is suited to the distributed storage or transmission of information. On the other hand, each subinformation requires the same number of bits as the original information X, which is very inefficient from the viewpoint of the coding efficiency. This paper extends the (k, n) threshold scheme and proposes the (k, L, n) threshold scheme. In the proposed scheme, the original information can be recovered completely from any k subinformation, but no information concerning X is obtained at all from any $(k - L)$ subinformation. From any $(k - t)$ subinformation $(1 \leq t \leq L - 1)$, the information obtained for X contains the ambiguity of (t/L) $H(X)$. In (k, L, n) scheme, the bit-length of each subinformation is $1/L$ of the information X, which is a coding with very high efficiency. This paper presents a construction method for (k, L, n) threshold scheme, together with the discussion of its characteristics.

1. Introduction

As a method to protect the information from an illegal listener, the (k, n) threshold scheme was proposed by Shamir [1]. Recently, numerous studies have been made on its realization and applications [2 ~ 11]. The (k, n) threshold scheme is a method in which the information X^N is partitioned and coded into n subinformation to be stored or transmitted. The secret protection ability is as follows. If k subinformation among n subinformation is obtained, the information X^N can be recovered completely, while no information concerning X^N can be obtained from any $(k, 1)$ subinformation.

Thus, when the (k, n) threshold scheme is employed, the information concerning X^N is not betrayed, even if up to $(k - 1)$ subinformation is revealed. Even if up to $(n - k)$ subinformation is destroyed, X^N can be recovered completely. Consequently, it is suited to the distributed storage of the secret key and the transmission of secret information utilizing more than one path. On the other hand, each subinformation in the (k, n) threshold scheme requires the same bit-length as the original information X^N [3] which is very inefficient from the viewpoint of the coding efficiency.

This paper proposes the (k, L, n) threshold scheme, which is an extension of the (k, n) threshold scheme, presenting the coding for its realization. In the (k, L, n) scheme, the original information can be recovered completely, if any k out of n subinformation is obtained, but no information can be obtained at all concerning X^N from any $(k - L)$ subinformation. If any $(k - t)$ subinformation is obtained $(1 \leq t \leq L - 1)$, the information concerning X^N is obtained to some extent with the decrease of n.

Using the (k, L, n) scheme, each subinformation requires the bit-length of $1/L$ compared with the original information X^N, which is a great improvement over the (k, n) scheme. For example, consider the case where $n = 10$ and the threshold is to be set approximately half of n. Using the (6, 2, 10) threshold scheme, the bit-length of the subinformation can be halved compared with the (5, 10) or (6, 10) threshold scheme, while realizing the similar extent of secret protection characteristics.

In Sect. 2, the (k, L, n) code is defined as the code realizing the (k, L, n) threshold scheme, discussing its characteristics. Section 3 presents the construction method for the (k, L, n) code, applying the coding by Karmin et al. [3] which realizes the (k, n) threshold scheme. The secret protection performance is evaluated as in

ISSN8756-6621/86/0009-0046$7.50/0

the past (k, n) threshold scheme, by the entropy function, assuming that the information source is a memoryless uniform probability source. Section 4 shows that the (k, L, n) code can be realized using the residue operation modulo 2^M, which is suited to the general-purpose computer.

Reference [12] presented a coding theorem for the more general case for $n = 2$ and 3, where the output symbols of the information source are not of uniform probability. Reference [5] presented an extended (k, n) threshold scheme using the Stone code [13], which is essentially equivalent to the (k, L, n) threshold scheme from the viewpoint of code construction. The difference is that the extension in [5] is made aiming primarily at the error correction.

2. (k, L, n) Threshold Scheme and (k, L, n) Code

Consider a discrete memoryless information source, where the output X_j $(j = 0, \pm1, \pm2, \cdots)$ takes the values of a finite discrete set $\mathcal{X}$. The number of elements in $\mathcal{X}$ is denoted by $|\mathcal{X}| = q$, where q is assumed for simplicity, as a prime number or its power. The symbols of $\mathcal{X}$ are assumed to occur with uniform probability. In other words, the following condition is satisfied by the information source for $j = 0, \pm1, \pm2, \cdots$:

$$P_r\{X_j = x_i\} = \frac{1}{q}, \quad x_i \in \mathcal{X}, \quad i = 1, 2, \cdots, q \tag{1}$$

$$H(X_j) = \log q \tag{2}$$

For this information source, the code (k, n) is defined as follows, where $N = mL$ $(1 \le L \le k)$:

$$f : \mathcal{X}^N \rightarrow \prod_{j=1}^{n} \mathcal{W}_j^m \tag{3}$$

ϕ: For any tuple of k of W_j^m,

$$\prod_{t=1}^{k} \mathcal{W}_{j_t}^m \rightarrow \mathcal{X}^N \tag{4}$$

In other words, $(W_1^m, W_2^m, \cdots, W_n^m) = f(X^N)$*, and for any tuple of k subinformation (simply called code words in the following) $(W_{j1}^m, W_{j2}^m, \cdots, W_{jk}^m)$, there holds $X^N = (W_{j1}^m, W_{j2}^m, \cdots, W_{jk}^m)$. Each symbol W_{ji} $(j = 1, 2, \cdots, n)$ of the code word $W_j^m = (W_{j1}, W_{j2}, \cdots, W_{jm})$, $j = 1, 2, \cdots$, takes the value in the set $\mathcal{W}_j$, where $|\mathcal{W}_j| = q$

By Eq. (3), the efficiency R_j of the code word W_j^m $(j = 1, 2, \cdots, n)$ is the same and is given by

$$R_j = \frac{N}{m} = L, \quad j = 1, 2, \cdots, n \tag{5}$$

The secret protection ability is evaluated based on the following (k, L, n) threshold condition.

Condition 1. For t in the range $0 \le t \le L$, any tuple of $(k - t)$ code words $(W_{j1}^m, W_{j2}^m, \cdots, W_{jk-t}^m)$ satisfies the following relation:

$$H(X^N \mid W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) = \frac{t}{L} H(X^N) = tm \log q \tag{6}$$

Condition 2. For t in the range $1 \le t \le L$, any tuple of $(k - t)$ code words $(W_{j1}^m, W_{j2}^m, \cdots, W_{jk-t}^m)$ and any t tuple $(X_{s1}^m, X_{s2}^m, \cdots, X_{st}^m)$ out of $X^N = (X_1^m, X_2^m, \cdots, X_L^m)$ satisfy the following relation:

$$H(X_{s_1}^m, X_{s_2}^m, \cdots, X_{s_t}^m \mid W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) = \frac{t}{L} H(X^N) = tm \log q \tag{7}$$

The code satisfying the condition 1 (f, ϕ) is called (k, L, n) code. The (k, L, n) code satisfying both conditions 1 and 2 is called the strong (k, L, n) code and the (k, L, n) code satisfying only condition 1 is called the weak (k, L, n) code.

Note 1. It is seen by setting $t = 0$ in Eq. (6) that X^N can be recovered completely from any k code words by using the (k, L, n) code. It is also seen by setting $t = L$ that no information at all can be obtained concerning X^N from any $(k - L)$ code words.

Note 2. The $(k, 1, n)$ code which is the case of $L = 1$, is the k-out-of-n code [3], realizing the (k, n) threshold scheme.

Note 3. It is seen from Eq. (5), that the code efficiency is better as L is increased, which represents the threshold of the secret protection performance.

As an example, compare the $(k, 1, n)$ code (C_1), $(k - 1, 1, n)$ code (C_2) and $(k, 2, n)$ code (C_3) for $q = 2$ and $N = 100$.

*The vector in this paper is always a row vector.

(number of bits in code word)

$C_1 : 100 \quad C_2 : 100 \quad C_3 : 50$

(secret protection ability): ambiguity $H(X^N)$

number of code words being k or more:

$C_1 : 1 \quad C_2 : 1 \quad C_3 : 1$

number of code words being $k - 1$:

$C_1 : 2^{100} (\approx 10^{30}) \quad C_2 : 1 \quad C_3 : 2^{50} (\approx 10^{15})$

number of code words being $k - 2$ or less:

$C_1 : 2^{100} \quad C_2 : 2^{100} \quad C_3 : 2^{100}$

As is seen from the above example, when $n >> L$, the (k, L, n) threshold scheme can decrease the length of the code word to $1/L$, while maintaining the similar degree of secret protection ability as that of the (k, n) threshold scheme. As in the above example, when the ambiguity of the order of 250 is sufficient for secret protection, the $(k, 2, n)$ code has essentially the same secret protection ability as that of the $(k, 1, n)$ code. In other words, when q^N/L is sufficiently large, the (k, L, n) threshold scheme can realize essentially the same secret protection ability as that of the (k, n) threshold scheme by $1/L$ code word length.

Note 4. In the weak (k, L, n) code, there is a possibility that a part of X^N can be recovered completely from $(k - t)$ code words where $t < L$. In the strong (k, L, n) code, the ambiguity of $(t/L)H(X^N)$ is maintained for any part of X^N, as is seen from Eq. (7). Consequently, the strong (k, L, n) code is more desirable than the weak (k, L, n) code. However, in the case where X^N does not have a meaning unless all of its information is recovered, the weak (k, L, n) code suffices.

It is then shown in the following that the above (k, L, n) code is the optimum in the sense that the code (f, ϕ) cannot achieve a larger ambiguity than Eqs. (6) and (7).

Theorem 1. If a code (f, ϕ) satisfies Eq. (6) for $t = 0$, any tuple of $(k - t)$ code words $(W_{j1}{}^m, W_{j2}{}^m, \cdots, W_{jk-t}{}^m)$ satisfies the following relation for $1 \le t \le L$:

$$H(X^N | W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \le (t/L) H(X^N) \tag{8}$$

Proof. The following relation applies for t in $1 \le t \le L$:

$$\begin{aligned} & H(X^N | W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ &= H(X^N, W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ &\quad - H(W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ &\ge H(X^N, W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t-1}}^m) \\ &\quad - H(W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t-1}}^m) \\ &\quad - H(W_{j_{k-t}}^m | W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t-1}}^m) \\ &= H(X^N | W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t-1}}^m) \\ &\quad - H(W_{j_{k-t}}^m | W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t-1}}^m) \\ &\ge H(X^N | W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t-1}}^m) - H(W_{j_{k-t}}^m) \\ &\ge H(X^N | W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t-1}}^m) - m \log q \end{aligned} \tag{9}$$

Consequently, if

$$H(X^N | W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_k}^m) = 0 \tag{10}$$

the following relation holds for t in $1 \le t \le L$:

$$\begin{aligned} & H(X^N | W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ &\le H(X^N | W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_k}^m) + t m \log q \\ &= t m \log q = (t/L) H(X^N) \end{aligned} \tag{11}$$

(End of Proof.)

It is obvious from Eq. (8) that

$$\begin{aligned} & H(X_{s_1}^m, X_{s_2}^m, \cdots, X_{s_t}^m | W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ &\le (t/L) H(X^N) \end{aligned} \tag{12}$$

also holds for the condition 2.

3. Construction of (k, L, n) Code

The k-out-of-n code encoding L information at the same time [3] can be utilized in the construction of the strong (k, L, n) code.* Consider the following code. Let A_S $(s = 1, 2, \cdots, L)$ and B_j $(j = 1, 2, \cdots, n)$ be $km \times km$ matrices on $GF(q)$. For given $X^N = (X_1{}^M, X_2{}^m, \cdots, X_L{}^m)$, the km-dimensional vector U^{km} is determined at random, satisfying

$$X_s^m = U^{km} A_s, \quad s = 1, 2, \cdots, L \tag{13}$$

Then the code word $W_j{}^m$ is determined by

$$W_j^m = U^{km} B_j, \quad j = 1, 2, \cdots n \tag{14}$$

The matrices A_S $(s = 1, 2, \cdots, L)$ and B_j $(j = 1, 2, \cdots, n)$ are made public. Letting

$$G = [A_1, A_2, \cdots, A_L, B_1, B_2, \cdots, B_n] \tag{15}$$

One can write that

$$(X_1^m, X_2^m, \cdots, X_L^m, W_1^m, W_2^m, \cdots, W_n^m) = U^{km} G \tag{16}$$

---

*Reference [3] showed that the conditions 1 and 2 are satisfied when $t = 0$ and $t = 1$.

This is called the *UG* code. The following theorem applies to the *UG* code.

Theorem 2. The necessary and sufficient condition for the *UG* code to be a strong (k, L, n) code is that any k matrices $\{C_{j1}, C_{j2}, \cdots, C_{jk}\}$ chosen from $\{A_1, A_2, \cdots, A_L, B_1, B_2, \cdots, B_n\}$ satisfy

$$\operatorname{rank}[C_{j_1}, C_{j_2}, \cdots, C_{j_k}] = km \tag{17}$$

The necessary and sufficient condition for *UG* code to be a weak (k, L, n) code is that any k matrices $\{B_{j1}, B_{j2}, \cdots, B_{jk}\}$ chosen from $\{B_1, B_2, \cdots, B_n\}$ satisfy

$$\operatorname{rank}[B_{j_1}, B_{j_2}, \cdots, B_{j_k}] = km \tag{18}$$

and any $k - L$ matrices $\{B_{j1}, B_{j2}, \cdots, B_{jk-L}\}$ chosen from $\{B_1, B_2, \cdots, B_n\}$ satisfy

$$\operatorname{rank}[A_1, A_2, \cdots, A_L, B_{j_1}, B_{j_2}, \cdots, B_{j_{k-L}}] = km \tag{19}$$

Proof. It is shown first that the condition 1 holds when Eqs. (18) and (19) are satisfied. Since any k matrices $\{B_{j1}, B_{j2}, \cdots, B_{jk}\}$ satisfy Eq. (18), U^{km} is determined by

$$U^{km} = [W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_k}^m][B_{j_1}, B_{j_2}, \cdots, B_{j_k}]^{-1} \tag{20}$$

and X^N is determined by Eq. (13). Consequently, Eq. (6) applies for $t = 0$.

Then consider the case of $1 \le t \le k$. The number of U^{km} satisfying Eq. (13) is $q^{(k-L)m}$ for each $x^N \in \mathcal{X}^N$, which are chosen at random with uniform probability. Consequently,

$$H(U^{km} \mid X^N) = \sum_{x^N} P_r(x^N) H(U^{km} \mid x^N) = (k-L)m \log q$$

Since X^N is determined uniquely from U^{km} by Eq. (13),

$$\begin{aligned} H(U^{km}) &= H(U^{km} X^N) = H(X^N) + H(U^{km} \mid X^N) \\ &= H(X^N) + (k-L)m \log q \end{aligned} \tag{21}$$

From the forementioned relation, the following is obtained:

$$\left.\begin{aligned} & H(X^N \mid W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ & \quad \ge I(X^N; U^{km} \mid W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ & \quad = H(U^{km} \mid W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ & \quad\quad - H(U^{km} \mid X^N, W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ (*1) \quad & \quad = H(U^{km} \mid W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ & \quad \ge H(U^{km}) - \sum_{s=1}^{k-t} H(W_{j_s}^m) \\ & \quad \ge H(U^{km}) - (k-t)m \log q \\ (*2) \quad & \quad \ge H(X^N) + (L-t)m \log q \\ & \quad = tm \log q \\ & \quad = (t/L) H(X^N) \end{aligned}\right\} \tag{22}$$

where (*1) is due to Eq. (18) and (*2) is due to Eq. (21). Equation (6) follows from Eqs. (8) and (22).

Similarly, the following relation applies when Eq. (17) is satisfied:

$$\left.\begin{aligned} & H(X_{s_1}^m, X_{s_2}^m, \cdots, X_{s_t}^m \mid W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ & \quad \ge I(X_{s_1}^m, X_{s_2}^m, \cdots, X_{s_t}^m; U^{km} \mid W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ & \quad = H(U^{km} \mid W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ & \quad\quad - H(U^{km} \mid X_{s_1}^m, X_{s_2}^m, \cdots, X_{s_t}^m, W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ (*3) \quad & \quad = H(U^{km} \mid W_{j_1}^m, W_{j_2}^m, \cdots, W_{j_{k-t}}^m) \\ & \quad \ge tm \log q \\ & \quad = (t/L) H(X^N) \end{aligned}\right\} \tag{23}$$

where (*3) is due to Eq. (17) and Eq. (7) follows from Eqs. (12) and (23).

Conversely, the following can easily be shown when Eqs. (18) and (19) do not apply, there exists $(W_{j1}^m, W_{j2}^m, \cdots, W_{jk-t}^m)$ which does not satisfy Eq. (6). When Eq. (17) does not apply, there exists $(X_{s1}^m, X_{s2}^m, \cdots, X_{st}^m)$ which does not satisfy Eq. (7). (End of Proof.)

The following theorem holds concerning the choice of matrices $\{A_1, A_2, \cdots, A_L, B_1, B_2, \cdots, B_{k-L}\}$.

Theorem 3. Without loss of generality, one can write

$$A_s = \begin{bmatrix} 0_m \\ \vdots \\ 0_m \\ I_m \\ 0_m \\ \vdots \\ 0_m \end{bmatrix} \begin{matrix} \\ \\ \\ \} s-1 \\ \\ \\ \\ \end{matrix}, \qquad B_j = \begin{bmatrix} 0_m \\ \vdots \\ 0_m \\ I_m \\ 0_m \\ \vdots \\ 0_m \end{bmatrix} \begin{matrix} \\ \\ \\ \} L+j-1 \\ \\ \\ \\ \end{matrix}$$

$$s = 1, 2, \cdots, L \qquad j = 1, 2, \cdots, k-L \tag{24}$$

where O_m is the $m \times m$ zero matrix, and I_m is the $m \times m$ unit matrix.

Proof. Omitted (it is the same as that of Theorem 3 in [3]).

It follows from Theorem 3 that U^{km} can be written as $U^{km} = (X_1^m, X_2^m, \cdots, X_L^m, Z_1^m,$

Z_{2^m}, ···, Z_{K-L^m}), where Z_{jl}, l = 1, 2, ···, m of Z_{j^m} = (Z_{j1}, Z_{j2}, ···, Z_{jm}), j = 1, 2, ···, k - L) is a uniform random variable on GF(q). It is seen from Theorem 2 that it suffices to determine the (k, L, n) code, that the tuple of matrices {A_1, A_2, ···, A_L, B_1, B_2, ···, B_n} satisfying Eq. (17) or (18), (19) be determined. Given k, L, q and m, let the maximum possible value of n be n_{max}. Then by Theorem 4 in [3], n_{max} satisfies the following theorem concerning the strong (k, L, n) code.

Theorem 4. When a strong (k, L, n) code is constructed using UG code, n_{max} satisfies the following relations:

$$\text{if } q^m > k, \qquad q^m - L + 1 \leq n_{max} \leq q^m + k - L - 1 \tag{25}$$

$$\text{if } q^m \leq k, \qquad n_{max} = k - L + 1 \tag{26}$$

Proof. Omitted (it is obvious from Theorem 4 in [3]).

The lower bound for n_{max} given by Eq. (26) can be achieved by determining G = [A_1, A_2, ···, A_L, B_1, B_2, ···, B_n] as in Eq. (27),* where m is set as I. When m > 1, βI_m can be used instead of each element β (= 0, 1, α, α^2, ···, α^{q-1}) in Eq. (27); α is a primitive element of GF(q):

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 & \cdots\cdots & 1 \\ 0 & 0 & \alpha & \alpha^2 & \cdots\cdots & \alpha^{q-1} \\ 0 & 0 & \alpha^2 & \alpha^4 & \cdots\cdots & \alpha^{(q-1)2} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ 0 & 1 & \alpha^{k-1} & \alpha^{2(k-1)} & \cdots\cdots & \alpha^{(q-1)(k-1)} \end{bmatrix} \tag{27}$$

The following bound exists for the weak (k, L, n) code.

Theorem 5. When a weak (k, L, n) code is constructed using the UG code, n_{max} satisfies the following relation:

$L = k$,

$$\text{if } q^m > k, \quad q^m + 1 \leq n_{max} \leq q^m + k - 1 \tag{28}$$

$$\text{if } q^m \leq k, \qquad n_{max} = k + 1 \tag{29}$$

$L = k - 1$,

$$\text{if } q^m > k, \quad q^m \leq n_{max} \leq q^m + k - 1 \tag{30}$$

$q^m \leq k$,

$$\text{if } q^m \neq 2 \qquad n_{max} = k + 1 \tag{31}$$

$$\text{if } q^m = 2, \text{ and } k \text{ is odd} \qquad n_{max} = k + 1 \tag{32}$$

$$\text{if } q^m = 2 \text{ and } k \text{ is even} \qquad n_{max} = k \tag{33}$$

$1 < L < k - 1$,

$$\text{if } q^m > k, \quad q^m + 1 \leq n_{max} \leq q^m + k - 1 \tag{34}$$

$$\text{if } k - 1 \leq q^m \leq k, \quad k \leq n_{max} \leq k + 1 \tag{35}$$

$$\text{if } \max(L, k - L) < q^m < k - 1, \; n_{max} \leq k + 1 \tag{36}$$

$$\text{if } q^m \leq \max(L, k - L), \text{ there does not exist}$$

$L = 1$,

$$\text{if } q^m > k, \quad q^m + 1 \leq n_{max} \leq q^m + k - 1 \tag{37}$$

$$\text{if } q^m \leq k, \quad n_{max} = k \tag{38}$$

Proof. It is shown in the Appendix.

4. (k, L, n) Code and Pseudo (k, L, n) Code by Residue Operation Modulo 2^M

The (k, L, n) code can be constructed using Eq. (27) and matrices of Eqs. (A3) to (A9). However, the operations must always be made on GF(q), which is fairly complicated when N is large. In the (k, L, n) code, nCk decoders must be prepared, and consequently, it is desirable that the encoding and decoding operations should be as simple as possible.

From such a viewpoint, this section discusses the construction of (k, L, n) code using only the residue operation modulo 2^M which is practically important and can easily be calculated by the general purpose computer. It is set in the following that m = 1 for simplicity. The reasoning for the case of m > 1 is similar.

As an example, consider the simplest case of (2, 1, 2) code, i.e., 2-out-of-2 code. By Theorems 3 and 4, the matrix G is set as

$$G = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \tag{39}$$

and U = (X, Z) can be used as U. The information X is $X \in$ GF(2^M) and Z is a uniform random variable on GF(2^M). Then the coding is made as follows:

$$W_1 = Z \tag{40}$$

*The UG code which can be constructed by Eq. (27) includes the code by Shamir [1] by polynomial interpolation [3, 6].

$$W_2 = X + Z \tag{41}$$

where "+" is the addition on GF(2^M). The decoding is made by

$$X = W_2 - W_1 \tag{42}$$

where "-" is the subtraction on GF(2^M).

As is seen in this example, only additions and subtractions are used when the matrix G is composed only of elements {0, 1}. Thus, the additive group suffices, not the field. Using the residue operation modulo 2^M, which composes an additive group, the (2, 1, 2) code can be constructed as follows where X, $Z \in \mathcal{I}_{2^M} \triangleq \{0, 1, 2, \cdots, 2^M - 1\}$ and Z is a uniform random variable on $\mathcal{I}_{2^M}$:

encoding:
$$\begin{cases} W_1 = Z & (43) \\ W_2 = (X + Z) \bmod 2^M & (44) \end{cases}$$

decoding:
$$X = (W_2 - W_1) \bmod 2^M \tag{45}$$

For a strong or weak (k, L, n) code to exist by the residue operation modulo 2^M, it suffices that there exists a matrix G satisfying Eq. (17), (18), or (19) on GF(2).

For the strong (k, L, n) code, there must hold

$$k \leq n_{\max} = k - L + 1 \tag{46}$$

by Eq. (26) of Theorem 4. Consequently, the code exists when

$$n = k, \quad L = 1 \tag{47}$$

By Theorem 5, the weak (k, L, n) code exists for the following cases:

$$\text{if } L = 1, \ 2 \leq k, \ n = k \tag{48}$$

$$\text{if } L = k - 1 \text{ is even}, \quad n = k, k + 1 \tag{49}$$

$$\text{if } L = k - 1 \text{ is odd}, \quad n = k \tag{50}$$

$$\text{if } L = k \geq 2, \ n = k, k + 1 \tag{51}$$

Thus, for (k, L, n) satisfying Eqs. (47) to (51), the (k, L, n) code can be constructed by the residue operation modulo 2^M. For other combinations of (k, L, n), the construction of the (k, L, n) code by residue operation modulo 2^M may be impossible. For (k, L, n) other than those satisfying Eqs. (47) to (51), however, the code which almost satisfies the condition for the (k, L, n) code can be constructed as follows, using the residue operation modulo 2 .

When a residue operation modulo 2^M is performed, the inverse to the multiplication cannot be determined uniquely. However, when a, c ($\in \mathcal{I}_{2^M}$) are given in

$$a \cdot b \mod 2^M = c \tag{52}$$

the number of b's satisfying Eq. (52) is a or less. Using a matrix G which has elements other than {0, 1}, as small natural numbers as possible and satisfies Eq. (17), (18) or (19), it is expected that a code which almost satisfies the condition for the (k, L, n) code (i.e., Eq. (6) or (7)) can be constructed by using the residue operation modulo 2^M.

As an example, consider the (2, 1, 3) code, i.e., 2-out-of-3 code. The following matrices are used as G and U:

$$G = \begin{bmatrix} 1 & 0 & 1 & 1 \\ 0 & 1 & 1 & 2 \end{bmatrix} \tag{53}$$

$$U = [X, Z] \tag{54}$$

The information X is $X \in \mathcal{I}_{2^M}$ and Z is a uniform random variable on $\mathcal{I}_{2^M}$. Then the coding is made as follows:

$$W_1 = Z \tag{55}$$

$$W_2 = (X + Z) \bmod 2^M \tag{56}$$

$$W_3 = (X + 2Z) \bmod 2^M \tag{57}$$

This code has the following ambiguity:

$$H(X|W_1) = H(X|W_2) = M \tag{58}$$

$$H(X|W_3) = M - 1 \tag{59}$$

$$H(X|W_i W_j) = 0 \quad i \neq j, \quad i, j = 1, 2, 3 \tag{60}$$

Except for Eq. (59), the code satisfies the conditions for (2, 1, 3) code. $H(X|W3)$ is less by one than M, but the secret protection ability is almost the same as that of the (2, 1, 3) code, when M is large. Such a code, which does not completely satisfy the condition for the (k, L, n) code, but practically has the same ability as that of the (k, L, n) code, is called the psuedo (k, L, n) code.

The pseudo (k, L, n) code can be constructed by the residue operation modulo 2^M for combinations of (k, L, n) other than those satisfying Eqs. (47) to (51). Depending on the choice of the matrix G, however, it may happen that Eq. (6) of condition 1 is not completely satisfied for $t = 0$. In such a case, X^N cannot be determined uniquely by a certain set of k code words. In order that X^N can always be recovered by any k code words, one must be careful about the choice of G.

In the following, the matrix G for k equal to or less than 5 is shown for the (k, L, n) node and the pseudo (k, L, n) code obtained by the residue operation modulo 2^M. For the pseudo (k, L, n) code, G is chosen so that the information is recovered uniquely by any k code words. {2, 3} in addition to {0, 1} is permitted as the element of the matrix. GkL^S is for the strong (k, L, n) code, and GkL^W is for the weak (k, L, n) code. The length of n applicable to each GkL is also shown. The n marked by asterisk produces a pseudo (k, L, n) code.

When a (k, L, n) code is constructed from these matrices, the code word is determined by

$$[W_1, W_2, \cdots, W_n] = [X_1, X_2, \cdots, X_L, Z_1, Z_2, \cdots, Z_{k-L}]\, g \quad (61)$$

where g is the submatrix from the $(L+1)$th to the $(L+n)$th column of GkL. The addition is the residue modulo 2^M. X_i, Z_i $\in \mathcal{S}_{2^M}$ and Z is a uniform random variable on $\mathcal{S}_{2^M}$:

$$G_{21}^S = \begin{bmatrix} 1&0&1&1 \\ 0&1&1&2 \end{bmatrix} \quad (n=2, 3^*)$$

$$G_{22}^S = \begin{bmatrix} 1&0&1&1&2 \\ 0&1&1&2&1 \end{bmatrix} \quad (n=2^*, 3^*)$$

$$G_{22}^W = \begin{bmatrix} 1&0&1&0&1 \\ 0&1&0&1&1 \end{bmatrix} \quad (n=2, 3)$$

$$G_{31}^S = \begin{bmatrix} 1&0&0&1 \\ 0&1&0&1 \\ 0&0&1&1 \end{bmatrix} \quad (n=3)$$

$$G_{32}^S = \begin{bmatrix} 1&0&0&1&1 \\ 0&1&0&1&2 \\ 0&0&1&1&3 \end{bmatrix} \quad (n=3^*)$$

$$G_{33}^S = \begin{bmatrix} 1&0&0&1&1&1 \\ 0&1&0&1&2&3 \\ 0&0&1&1&3&2 \end{bmatrix} \quad (n=3^*)$$

$$G_{32}^W = \begin{bmatrix} 1&0&1&0&0&1 \\ 0&1&0&1&0&1 \\ 0&0&1&1&1&1 \end{bmatrix} \quad (n=3, 4)$$

$$G_{33}^W = \begin{bmatrix} 1&0&0&1&0&0&1 \\ 0&1&0&0&1&0&1 \\ 0&0&1&0&0&1&1 \end{bmatrix} \quad (n=3, 4)$$

$$G_{41}^S = \begin{bmatrix} 1&0&0&0&1 \\ 0&1&0&0&1 \\ 0&0&1&0&1 \\ 0&0&0&1&1 \end{bmatrix} \quad (n=4)$$

$$G_{42}^W = \begin{bmatrix} 1&0&0&0&1&1 \\ 0&1&0&0&1&2 \\ 0&0&1&0&1&1 \\ 0&0&0&1&1&2 \end{bmatrix} \quad (n=4^*)$$

$$G_{43}^W = \begin{bmatrix} 1&0&0&1&0&0&0&1 \\ 0&1&0&0&1&0&0&1 \\ 0&0&1&0&0&1&0&1 \\ 0&0&0&1&1&1&1&2 \end{bmatrix} \quad (n=4, 5^*)$$

$$G_{44}^W = \begin{bmatrix} 1&0&0&0&1&0&0&0&1 \\ 0&1&0&0&0&1&0&0&1 \\ 0&0&1&0&0&0&1&0&1 \\ 0&0&0&1&0&0&0&1&1 \end{bmatrix} \quad (n=4, 5)$$

$$G_{51}^S = \begin{bmatrix} 1&0&0&0&0&1 \\ 0&1&0&0&0&1 \\ 0&0&1&0&0&1 \\ 0&0&0&1&0&1 \\ 0&0&0&0&1&1 \end{bmatrix} \quad (n=5)$$

$$G_{52}^W = \begin{bmatrix} 1&0&0&0&0&1&1&2 \\ 0&1&0&0&0&1&2&1 \\ 0&0&1&0&0&1&1&3 \\ 0&0&0&1&0&1&2&2 \\ 0&0&0&0&1&1&3&1 \end{bmatrix} \quad (n=5^*, 6^*)$$

$$G_{53}^W = \begin{bmatrix} 1&0&0&0&0&1&1&0 \\ 0&1&0&0&0&1&0&1 \\ 0&0&1&0&0&1&0&0 \\ 0&0&0&1&0&1&1&1 \\ 0&0&0&0&1&1&2&2 \end{bmatrix} \quad (n=5^*)$$

$$G_{54}^W = \begin{bmatrix} 1&0&0&0&1&0&0&0&0&1 \\ 0&1&0&0&0&1&0&0&0&1 \\ 0&0&1&0&0&0&1&0&0&1 \\ 0&0&0&1&0&0&0&1&0&1 \\ 0&0&0&0&1&1&1&1&1&1 \end{bmatrix} \quad (n=5, 6)$$

$$G_{55}^W = \begin{bmatrix} 1&0&0&0&0&1&0&0&0&0&1 \\ 0&1&0&0&0&0&1&0&0&0&1 \\ 0&0&1&0&0&0&0&1&0&0&1 \\ 0&0&0&1&0&0&0&0&1&0&1 \\ 0&0&0&0&1&0&0&0&0&1&1 \end{bmatrix} \quad (n=5, 6)$$

5. Conclusions

This paper proposed the (k, L, n) threshold scheme which is an extension of

the (k, n) threshold scheme suited to the distributed storage or transmission of the information. The method of constructing the (k, L, n) code is also presented to realize the proposed scheme. Using the (k, L, n) threshold scheme, the bit length of the subinformation can be decreased to $1/L$, while maintaining practically the same secret protection ability as that of the (k, n) scheme. Thus, a very efficient coding is realized. The encoding and the decoding of the (k, L, n) code are of the same extent of complexity as those of the k-out-of-n code realizing the (k, n) threshold scheme. Consequently, the (k, L, n) threshold scheme will be applied directly to any field where the application of the (k, n) threshold scheme is now considered.

REFERENCES

1. A. Shamir. How to share a secret, Commun., ACM, 22, pp. 612-613 (Nov. 1979).
2. R.J. McEliece and D.V. Sarwate. On sharing secrets and Reed-Solomon codes, Commun., ACM, 24, pp. 583-584 (Sept. 1981).
3. E.D. Karnin, J.W. Greene and M.E. Hellman. On secret sharing systems, I.E.E.E. Trans. Inf. Theory, IT-29, 1, pp. 35-41 (Jan. 1983).
4. C. Asmuth and J. Bloom. A modular approach to key safeguarding, I.E.E.E. Trans. Inf. Theory, IT-29, 2, pp. 208-210 (Mar. 1983).
5. Matsumi, Tokiwa, Kasahara, Namekawa and Tanaka. A study of secret sharing systems using algebraic error-correcting codes, Tech. Rep. I.E.C.E., Japan, IT84-8 (May 1984).
6. Matsumi, Tokiwa, Kasahara and Namekawa. Notes on (K, N) threshold scheme, 67th Symp. on Inf. Theory and Its Appl., pp. 158-163 (Nov. 1984).
7. Yamaguchi and Imai. On stone code, Tech. Rep. I.E.C.E., Japan, IT84-27 (Sept. 1984).
8. Yamaguchi and Imai. A decoding method for stone codes and its applications, 7th Symp. on Inf. Theory and Appl., pp. 36-40 (Nov. 1984).
9. K. Koyama. Sharing Cryptographic Keys in Multigroup and Its Analysis, Trans. Inf. Proc. Soc. Jap., 22, 2, pp. 81-88 (Mar. 1981).
10. K. Koyama. Cryptographic key sharing methods for multigroups and security analysis, Trans. I.E.C.E., Japan (Section E), E66, 1, pp. 13-20 (Jan. 1983).
11. K. Matsui. File management system by k-out-of-N individual identification, Trans. Inf. Proc. Soc. Jap., 25, 3, pp. 351-356 (May 1984).
12. H. Yamamoto. On secret sharing communication systems with two or three channels, I.E.E.E. Trans. Inf. Theory (to be published).
13. J.J. Stone. Multiple-burst error correction with the Chinese remainder theorem, J. SIAM, 11, 1, pp. 74-81 (Mar. 1963).

APPENDIX

Proof of Theorem 5
(Upper bound for $n_{\max}$)

For Eq. (18) to hold, $n_{\max}$ must satisfy the following relation, as is seen from the proof for Corollary 1 of Theorem 4 [3]. In Theorem 4 in [3] $n_{\max} + 1$ should be replaced by $n_{\max}$.

$$\text{if} \quad q^m > k, \quad n_{\max} \leq q^m + k - 1 \tag{A1}$$

$$\text{if} \quad q^m \leq k, \quad n_{\max} \leq k + 1 \tag{A2}$$

(lower bound for $n_{\max}$)

The lower bound for $n_{\max}$ can be achieved by determining the matrix G as follows, where m is set as 1. For $m > 1$, it suffices to use βI_m instead of the elements β (= 0, 1, α, α^2, $\cdots$, α^{q-1}) of the matrix G:

$L = k$:

when $q^m > k$

$$G = \left[\begin{array}{ccc} & 0\ 1 & \\ & 0\ 0 & \\ I_k & \vdots\ \vdots & H_{00} \\ & 0\ 0 & \\ & 1\ 0 & \end{array}\right] \tag{A3}$$

where H_{st} is the following matrix:

$$H_{st} = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ \alpha & \alpha^2 & \cdots & \alpha^{q-1-t} \\ \alpha^2 & \alpha^4 & \cdots & \alpha^{(q-1-t)2} \\ \vdots & \vdots & & \\ \alpha^{k-1-s} & \alpha^{2(k-1-s)} & \cdots & \alpha^{(q-1-t)(k-1-s)} \end{bmatrix} \tag{A4}$$

if $q^m \leq k$

$$G = \left[\begin{array}{cc|c} & & 1 \\ I_k & I_k & 1 \\ & & \vdots \\ & & 1 \end{array}\right] \tag{A5}$$

$$L = k - 1,$$

if $q^m > k$

$$G = [\, I_k \quad H_{00} \,] \tag{A6}$$

if $q^m \leq k$

$$G = \left[\begin{array}{ccc} & & 0\ 1 \\ & & 0\ 1 \\ I_{k-1} & I_{k-1} & \\ & & 0\ 1 \\ 0\ 0 \cdots\cdots 0\ 1 & 1 \cdots\cdots 1\ 1 & X \end{array}\right] \tag{A7}$$

Here, x is an element of GF(q) other than 0 and 1 + 1 + ••• + 1 (L times addition of 1's). If q = 2 and k is even (L is odd), such an element x does not exist, but always exists for other cases:

$$1 < L < k - 1,$$

$$q^m > k$$

$$G[\, I_k \quad H_{00} \,] \tag{A8}$$

if $q^m = k$

$$G = \left[\begin{array}{c|cc} & & 0 \cdots\cdots 0 \\ I_L & I_L & \vdots \quad\quad \vdots \\ & & 0 \cdots\cdots 0 \\ \hline 0 \cdots\cdots 0 & 1\ 0 & \\ \vdots \quad\quad \vdots & 0\ 0 & \\ \vdots \quad\quad \vdots & \vdots\ \vdots & H_{L1} \\ \vdots \quad\quad \vdots & 0\ 0 & \\ 0 \cdots\cdots 0 & 0\ 1 & \end{array}\right] \tag{A9}$$

if $q^m = k - 1$

$$G = \left[\begin{array}{c|cc} & & 0 \cdots\cdots 0 \\ I_L & I_L & \vdots \quad\quad \vdots \\ & & 0 \cdots\cdots 0 \\ \hline 0 \cdots\cdots 0 & 1\ 0 & \\ \vdots \quad\quad \vdots & 0\ 0 & \\ \vdots \quad\quad \vdots & \vdots\ \vdots & H_{L0} \\ \vdots \quad\quad \vdots & 0\ 0 & \\ 0 \cdots\cdots 0 & 0\ 1 & \end{array}\right] \tag{A10}$$

It suffices to consider the following matrix, as is seen from Theorem 3:

$$G = \left[\begin{array}{c|c} I_L & A \\ \hline 0 \cdots\cdots 0 & \\ \vdots \quad\quad \vdots & B \\ 0 \cdots\cdots 0 & \end{array}\right] \tag{A11}$$

In order that Eq. (19) is satisfied, any (k - L) columns of submatrix B of Eq. (A11) must be linearly independent. By Theorem 3 of [3], on the other hand, the maximum number of linearly independent columns in B is k - L + 1 for $q^m \leq k - L$. Consequently,

$$n_{max} \leq k - L + 1 < k \tag{A12}$$

When $q^m \leq L$, the number maximum of linearly independent columns in B is $q^m + k - L - 1$ or less. Consequently,

$$n_{max} \leq q^m + k - L - 1 \leq k - 1 < k \tag{A13}$$

This contradicts

$$n_{max} \geq k \tag{A14}$$

Consequently, there does not exist a matrix G satisfying Eq. (19).

L = 1: In this case, the code coincides with the strong (k, L, n) code, and

$$\text{if} \quad q^m > k, \quad q^m \leq n_{max} \leq q^m + k - 2 \tag{A15}$$

$$\text{if} \quad q^m \leq k, \quad n_{max} = k \tag{A16}$$