# A Construction of Public-Key Cryptosystem Using Algebraic Coding on the Basis of Superimposition and Randomness

Masao Kasahara *

**Abstract**— In this paper, we present a new class of public-key cryptosystem (PKC) using algebraic coding on the basis of superimposition and randomness. The proposed PKC is featured by a generator matrix, in a characteristic form, where the generator matrix of an algebraic code is repeatedly used along with the generator matrix of a random code, as sub-matrices. This generator matrix, in the characteristic form, will be referred to as $K$-matrix. We show that the $K$-matrix yields the following advantages compared with the conventional schemes:

**(i)** It realizes an abundant supply of PKCs, yielding more secure PKCs.

**(ii)** It realizes a fast encryption and decryption process.

**Keywords:** algebraic coding, random coding, public-key cryptosystem

## 1 Introduction

Extensive studies have been made of the Public Key Cryptosystem (Hereinafter Public-Key Cryptosystem will be abbreviated as PKC). The security of most PKCs depends on the difficulty of discrete logarithm problem or factorization problem. Thus it is desired to investigate another classes of PKC that do not rely on the difficulty of these two problems.

In this paper, we shall present a new class of PKC using algebraic coding on the basis of superimposition and randomness. As is well known, one of the most important applications of the error correcting codes to the field of cryptography is due to McEliece [1], who applied the Goppa codes to PKC. However, almost all classes of McEliece-Type Cryptosystem use powerful codes of large minimum distances such as Goppa codes [1,2].

On the other hand, in step with the recent advancement of the various digital systems such as error-control systems, extension field $F_{2^m}$ has been extensively studied and has been widely used for realizing secure and reliable storage and transmission systems. Thus constructing a new class of PKC over $F_{2^m}$ would be desirable, because the various precious and abundant knowledges on software and hardware techniques are available when using $F_{2^m}$.

In this paper, we shall present a new class of PKC over $F_{2^m}$ using algebraic coding on the basis of superimposition [2] and randomness. The proposed PKC is characterized by a $(k+r)\times 2n$ matrix, $K$-matrix, that is constituted by two repeatedly used generator matrices for algebraic code and a generator matrix for random code, as shown below:

$$K = \begin{bmatrix} G_A & ; & G_A \\ & R & \end{bmatrix}, \tag{1}$$

where $G_A$ is a $k\times n$ generator matrix of a large class of algebraic code and $R$ is an $r\times 2n$ generator matrix of random code. In Eq. (1), the notation $[\mathbf{X};\mathbf{Y}]$ implies the concatenation of matrices $\mathbf{X}$ and $\mathbf{Y}$. It should be noted that the codes generated by $G_A$ is very general in a sense that they are not necessarily required to have a large minimum distance, in a sharp contrast with a powerful code such as the generalized BCH codes [2]. We assume that almost all elements of $R$ are generated in a random manner. The proposed PKC constructed on the basis of $K$-matrix given by Eq. (1) will be referred to as $\mathrm{K_r}$-PKC, where $r$ stands for the number of row vectors of $R$.

In this paper, we show that the $K$-matrix yields the following advantages compared with the conventional generator matrix:

**(i)** It realizes an abundant supply of PKCs, yielding more secure PKCs.

**(ii)** It realizes a fast encryption and decryption process.

## 2 Preliminaries

In this section, for simplicity, we present $\mathrm{K_r}$-PKC where no permutation matrix is used. In later section we shall present $\mathrm{K_r}$-PKC where permutation matrices are used.

### 2.1 Public and Secret Keys

Let us define the following $(k+r)\times 2n$ matrix over $F_{2^m}$:

$$M = SK, \tag{2}$$

where $S$ is the $(k+r)\times(k+r)$ non-singular matrix over $F_{2^m}$. In Eq. (2), $M$ is made public and $S$ is kept secret.

**[Subset of Keys]**

**Public Key :** $M$

**Secret Key :** $S$,$K$

Another set of public keys such as $e=3$, $G_{\mathrm{RS},D}(X)$ will be added later on.

Besides public key $M$ will be used in a permuted form.

* Faculty of Informatics, Osaka Gakuin University Kishibe-Minami, Suita-Shi, Osaka 564-8511 Japan

### 2.2 Matrix $R$

Let the matrix $R$ in Eq. (1) be denoted by

$$R = [R_L; R_R] = \begin{bmatrix} R_{1,1}, \cdots, R_{1,n} & ; & R_{1,n+1}, \cdots, R_{1,2n} \\ R_{r,1}, \cdots, R_{r,n} & & R_{r,n+1}, \cdots, R_{r,2n} \end{bmatrix}, \quad (3)$$

where we assume that all the elements are chosen in a random manner under Condition $C_1$ which will be given later.

Based on $R$, let us define the following $(k+r) \times n$ matrix $R^+$ as follows:

$$R^+ = R_L + R_R. \quad (4)$$

Let us define another matrix $R_S$ whose columns are constituted by the randomly chosen $r$ columns of $R^+$ as follows:

$$R_S = \begin{bmatrix} R^+_{1J_1} & , \cdots, & R^+_{1J_r} \\ \vdots & & \vdots \\ R^+_{rJ_1} & , \cdots, & R^+_{rJ_r} \end{bmatrix}, \quad (5)$$

where we assume that the relation $1 \leq J_1 < J_2 < \cdots < J_r \leq n$ holds and $R^+_{ij}$ is given by

$$R^+_{ij} = R_{ij} + R_{i,j+n}. \quad (6)$$

**Condition $C_1$ :** Among $\binom{n}{r}$ choices of $R_S$'s for a given $R$, there exist sufficiently large amount of non-singular matrices. □

**Remark 1 :** As we shall discuss in the later section, in $K_r$-PKC, $r$ should be chosen sufficiently small compared with $k$ and the relation $n \cong k$ holds. Evidently, the probability that the randomly chosen $R_S$ over $F_{2^m}$ is non-singular takes on sufficiently large value for $m \gtrsim 3$. Thus Condition $C_1$ is satisfied and the obtaining of non-singular matrix $R_S$ yields no obstacle when designing the proposed $K_r$-PKC. □

### 2.3 Encryption and Decryption Processes

For an easy understanding, we shall describe here the encoding and decoding processes in a similar manner as the describing of algorithms.

**[Encryption].**

Encryption process can be given as follows:

Let the message vector $\mathbf{m}$ over $F_{2^m}$ be denoted as follows:

$$\mathbf{m} = (m_1, m_2, \cdots, m_k, \cdots, m_{k+r}). \quad (7)$$

The cyphertext $\mathbf{C}$ is given by

$$\mathbf{C} = \mathbf{m}M + \mathbf{e}, \quad (8)$$

where $\mathbf{e}$ is a $2n$-symbols error vector over $F_{2^m}$ which is generated in a random manner at the sender. Let us denote the weight of the error vector $\mathbf{e}$, $w(\mathbf{e})$, by

$$w(\mathbf{e}) = 2n - l_{2n}, \quad (9)$$

where $l_{2n}$ is an appropriately chosen positive integer depending on the given $K$-matrix. We also denote $\mathbf{m}M$ as follows:

$$\mathbf{m}M = (m'_1, m'_2, \cdots, m'_k, \cdots, m'_{k+r}) \quad (10)$$

□

Let us partition the components of the error vector $\mathbf{e}$ as shown in Fig. 1.

$$\mathbf{e} = (\mathbf{x}_L; \mathbf{y}_L; \mathbf{y}_R; \mathbf{x}_R). \quad (11)$$

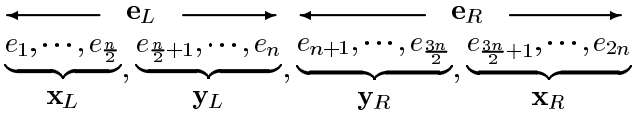


Fig. 1 Partition of components of $\mathbf{e}$.

In the following, we assume that the error vector $(\mathbf{y}_R, \mathbf{y}_L)$ is given so that it can be corrected by the code generated by the generator matrix $G_A$.

**[Decryption].**

**Step 1 :** Let us denote the received version of cyphertext $\mathbf{C}$ as follows:

$$\mathbf{C} = \tilde{\mathbf{m}}K + \mathbf{e}, \quad (12)$$

where $\tilde{\mathbf{m}}$ is given by

$$\tilde{\mathbf{m}} = \mathbf{m}S, \quad (13)$$

and is denoted as

$$\tilde{\mathbf{m}} = (\tilde{m}_1, \tilde{m}_2, \cdots, \tilde{m}_k, \cdots, \tilde{m}_{k+r}). \quad (14)$$

Let the error vector $\mathbf{e}$ be represented by

$$\mathbf{e} = (\mathbf{e}_L; \mathbf{e}_R), \quad (15)$$

where we let $\mathbf{e}_L$ and $\mathbf{e}_R$ be denoted by

$$\mathbf{e}_L = (e_1, e_2, \cdots, e_n) \quad (16)$$

and

$$\mathbf{e}_R = (e_{n+1}, e_{n+2}, \cdots, e_{2n}), \quad (17)$$

respectively. In a similar manner as the error vector $\mathbf{e}$, the $\tilde{\mathbf{m}}K$ is partitioned to $(\tilde{\mathbf{m}}K)_L$ and $(\tilde{\mathbf{m}}K)_R$ as follows:

$$\tilde{\mathbf{m}}K = \Big((\tilde{\mathbf{m}}K)_L; (\tilde{\mathbf{m}}K)_R\Big). \quad (18)$$

Note that $\tilde{\mathbf{C}}$ is now represented as

$$\tilde{\mathbf{C}} = \Big((\tilde{\mathbf{m}}K)_L + \mathbf{e}_L; (\tilde{\mathbf{m}}K)_R + \mathbf{e}_R\Big). \quad (19)$$

**Step 2 :** The message $(\tilde{m}_{k+1}, \cdots, \tilde{m}_{k+r})$ are decoded based on the following vector $\mathbf{v}$:

$$\mathbf{v} = (\tilde{\mathbf{m}}K)_L + \mathbf{e}_L + (\tilde{\mathbf{m}}K)_R + \mathbf{e}_R. \tag{20}$$

**Remark 2 :** In this Step 2, $\mathbf{e}_L + \mathbf{e}_R = (\mathbf{x}_L + \mathbf{y}_R; \mathbf{x}_R + \mathbf{y}_L)$ is decoded and is known to the receiver. In the following step, Step 4, $(\mathbf{x}_L + \mathbf{y}_R; \mathbf{0})$ will be used for decoding the remaining message $(\tilde{m}_1, \tilde{m}_2, \cdots, \tilde{m}_k)$. □

**Step 3 :** The decoded version, $(\hat{\tilde{m}}_{k+1}, \cdots, \hat{\tilde{m}}_{k+r})R$ is subtracted from the received cyphertext $\mathbf{C}$ given by Eq. (12). The message $(\tilde{m}_1, \tilde{m}_2, \cdots, \tilde{m}_k)$ is decoded by the following steps.

**Step 4 :** Let $\tilde{\mathbf{m}}'$ and $K'$ be given as follows:

$$\tilde{\mathbf{m}}' = (\tilde{m}_1, \tilde{m}_2, \cdots, \tilde{m}_k) \tag{21}$$

and

$$K' = [G_A; G_A]. \tag{22}$$

Then $\tilde{\mathbf{m}}'K'$ can be represented as

$$\tilde{\mathbf{m}}'K' = (\mathbf{F}_{G_L}; \mathbf{F}_{G_R}), \tag{23}$$

where $\mathbf{F}_{G_L}$ and $\mathbf{F}_{G_R}$ are given by

$$\mathbf{F}_{G_L} = \mathbf{F}_{G_R} = \tilde{\mathbf{m}}'G_A. \tag{24}$$

The syndrome $(\mathbf{x}_L + \mathbf{y}_R; \mathbf{0})$ obtained in Step 2 is added on $\mathbf{F}_G + (\mathbf{x}_L + \mathbf{y}_L)$ in the following manner:

$$\begin{aligned} &\mathbf{F}_{G_L} + (\mathbf{x}_L; \mathbf{y}_L) + (\mathbf{x}_L + \mathbf{y}_R; \mathbf{0}) \\ &\qquad = \mathbf{F}_{G_L} + (\mathbf{y}_R; \mathbf{y}_L), \end{aligned} \tag{25}$$

where $\mathbf{0}$ is the $n/2$-symbols vector of all zero elements. By correcting $(\mathbf{y}_R; \mathbf{y}_L)$, $\mathbf{F}_{G_L}$ is decoded, yielding the decoded message $(\hat{\tilde{m}}_1, \hat{\tilde{m}}_2, \cdots, \hat{\tilde{m}}_k)$.

**Step 5 :** The message vector $\hat{\mathbf{m}} = (\hat{m}_1, \hat{m}_2, \cdots, \hat{m}_{k+1}, \cdots, \hat{m}_{k+r})$ is decoded as follows:

$$\hat{\mathbf{m}} = \hat{\tilde{\mathbf{m}}} S^{-1}. \tag{26}$$

□

Let us denote the Hamming weight of the vector $\mathbf{e}_L + \mathbf{e}_R$ by

$$w(\mathbf{e}_L + \mathbf{e}_R) = n - l_n, \tag{27}$$

where $l_n$ is an appropriately chosen positive integer depending on the given $K$-matrix.

**Remark 3 :** It is evident that $l_n$ and $l_{2n}$ are required to be as small as possible in order to make our proposed PKC secure. We shall show, in the next section, that these values can be made sufficiently small. □

**Theorem 1 :** The messages $\tilde{m}_{k+1}, \cdots, \tilde{m}_{k+r}$ can be successfully decoded provided that there exists symbols with no error at the sufficient number of locations of the components of the vector $(\tilde{\mathbf{m}}K)_L + (\tilde{\mathbf{m}}K)_R$. □

# 3 Several Properties

## 3.1 Ambiguity of Matrix

When the following relation holds:

$$SG \neq G' \tag{28}$$

for any $K_0 \times K_0$ non-singular matrix $S$, then $K_0 \times N_0$ matrices, $G$ and $G'$, will be referred to as *different*.

**Definition 1 :** Let the number of different $G$'s that satisfy a given condition $C$ be represented as $\sharp\{G\}_C$. The ambiguity, $A_{GC}$, of the matrix $G$ that meets the given condition $C$ is defined as

$$A_{GC} = \log_2 \sharp\{G\}_C. \tag{29}$$

□

## 3.2 Several Parameters

The size of the public key, $S_{PK}$, is given by

$$S_{PK} = 2nm(k + r) \ \text{(bits)}. \tag{30}$$

The size of the cyphertext, $S_C$, is given by

$$S_C = 2nm \ \text{(bits)}. \tag{31}$$

The coding rate (information rate), $\rho$, is given by

$$\rho = \frac{k + r}{2n}. \tag{32}$$

Error rate $\varepsilon$ is given by

$$\varepsilon = \frac{w(\mathbf{e})}{2n}. \tag{33}$$

# 4 An Example of Vector y of Large Hamming Weight

Let us discuss here a method of the generating of a random error vector of large Hamming weight that can be successfully corrected at the receiver.

We assume that the generator matrix, $G$, is given as that of the quasi-cyclic code (qcc) over $F_{2^m}$ that is generated by the following polynomial $G(X)$:

$$G(X) = G_0 + G_1 X + \cdots + G_{g-1} X^{g-1} + X^g, \tag{34}$$

where $G_i \in F_{2^m}$ and $G_0 \neq 0$.

It is evident that the code generated by such a general polynomial has no systematic distance structure.

Let us denote $(\mathbf{y}_R, \mathbf{y}_L)$ by $\mathbf{y}$ and denote $\mathbf{y}$ by the following polynomial:

$$y(X) = y_R(X) X^{-n} + y_L(X), \tag{35}$$

where $y_R(X)$ and $y_L(X)$ stand for $\mathbf{y}_R$ and $\mathbf{y}_L$, respectively, and are denoted by

$$y_L(X) = e_{\frac{n}{2}+1} X^{\frac{n}{2}} + e_{\frac{n}{2}+1} X^{\frac{n}{2}+2} + \cdots + e_n X^{n-1} \tag{36}$$

and

$$y_R(X) = e_{n+1} X^n + \cdots + e_{\frac{3n}{2}} X^{\frac{3n}{2}-1}, \tag{37}$$

respectively.

In order to let the Hamming weight of vector $\mathbf{y}$ take on a large value, we construct $y(X)$ in the following form:

$$y(X) = \{\eta(X)\}^3, \tag{38}$$

where $\eta(X)$ is an error polynomial corresponding to the following error vector $\boldsymbol{\eta}$ randomly generated:

$$\boldsymbol{\eta} = (\eta_1, \eta_2, \cdots, \eta_{\frac{n-\varepsilon}{3}}), \quad \eta_i \in F_{2^m}, \tag{39}$$

where $\varepsilon$ is an appropriately chosen positive integer. Evidently, $\eta(X)$ can be decoded by the following equation:

$$\{F_{G_L}(X) + y(X)\}^{3^{-1}} \equiv \eta(X) \bmod G(X), \tag{40}$$

provided that the exponent of $G(X)$ is not divisible by 3.

It is evident that, once $\mathbf{y}_L(X) + \mathbf{y}_R(X)$ is decoded, each of the error vectors $(\mathbf{x}_L, \mathbf{0})$ and $(\mathbf{0}, \mathbf{x}_R)$ can be decoded by a single step of multiplication and division by the generator polynomial $G(X)$. Thus it is expected that the decoding process can be performed very fast, compared with the conventional method using Goppa codes etc.

Another class of generator polynomials will be given in Section 6.

In an example of $\mathrm{K_r}$-PKC given in the following section, we discuss the method of adding information symbols that can be considered sufficiently randomized, instead of an error vector.

**Remark 4 :** We assume that the message vector $\mathbf{m}$ over $F_{2^m}$ is sufficiently randomized, for example, by the using of common-key cryptosystem whose secret key are made public, yielding a sufficiently random message vector over $F_{2^m}$. □

We shall present here an example of the proposed PKC, $\mathrm{K_2}$-PKC assuming the error vector given by (38). In the following example, we assume that the $2r = 4$ locations of the codeword are required to be error free and the locations are made public.

**Example 1 :** Let the $K$-matrix of $\mathrm{K_2}$-PKC is given by

$$K = \begin{bmatrix} G & G \\ \mathbf{r}_L & \mathbf{r}_R \\ \mathbf{r}'_L & \mathbf{r}'_R \end{bmatrix}, \tag{41}$$

where $G$, $\mathbf{r}_L$, $\mathbf{r}_R$, $\mathbf{r}'_L$ and $\mathbf{r}'_R$ are given as follows:

$G$ **:** $127 \times 254$ matrix over $F_{2^7}$ that are generated by the polynomial $G(X)$ over $F_{2^7}$ whose degree is 127, given in Eq. (34).

$\mathbf{r}_L$, $\mathbf{r}_R$, $\mathbf{r}'_L$, $\mathbf{r}'_R$ **:** 254-symbols random vector over $F_{2^7}$.

The number of information symbols (in bits), $\kappa$, and the size of the cyphertext (in bits), $|C|$, are given by

$$\kappa = (k+2) \times m = 903 \ \text{(bits)} \quad \text{and} \quad |C| = 2n \times m = 3556 \ \text{(bits)}, \tag{42}$$

respectively.

The size of the public-key, $S_{pk}$, is given by

$$S_{pk} = (k+2) \times 2n \times m = 458724 \ \text{(bits)}. \tag{43}$$

The ambiguity of $K$ matrix [Eq. (41)] is given by

$$\begin{aligned} A_K &= \log_2 \sharp\{G\}_{qcc} + \log_2 \sharp\{(\mathbf{r}_L, \mathbf{r}_R)\}_R \\ &\quad + \log_2 \sharp\{(\mathbf{r}'_L, \mathbf{r}'_R)\}_R = 889 + 3556 \times 2 \\ &= 8001 \ \text{(bits)}, \end{aligned} \tag{44}$$

a sufficiently large value.

In Example 1, the error rate $\varepsilon$ takes on the value 0.992, extremely large value. If the message $\mathbf{m}'$s are added instead of error vectors in this example, the coding rates are given as 0.915 [see Eq. (50) given in the following section].

## 5 Discussions

### 5.1 Security Considerations and Improvements

**(I) Improvement using permutation matrix $P$**

So far we have discussed a class of $\mathrm{K_r}$-PKC that uses no permutation matrix. Evidently using no permutation matrix may weaken the proposed PKC. In this sub-section, we present a method for overcoming this problem. We first assume that the generator matrix $G_A$ in Eq. (1) generates the maximum distance separable code over $F_{2^m}$.

We assume here that the proposed $\mathrm{K_r}$-PKC uses a $(2n \times 2n)$ permutation matrix $P$ in a general form, yielding the following public key:

$$M = SKP \tag{45}$$

Using this public-key, it is easy to see that the $\mathrm{K_r}$-PKC generated by $G_A$ mentioned above is able to correct $n/4$ errors. In this case, error rate $\varepsilon$ is given by $\varepsilon = 1/4$, sufficiently large values when $k$ takes on the value $k \gtrsim 100$. Letting the weight of vector $(\mathbf{e}_L + \mathbf{e}_R)$ be denoted by $w(\mathbf{e}_L + \mathbf{e}_R)$, the error vector $\varepsilon_n$ in $\mathbf{v}$ is given by

$$\varepsilon_n = \frac{w(\mathbf{e}_L + \mathbf{e}_R)}{n} \tag{46}$$

It is easy to see that when the row vector of $R$ are sufficiently random, messages $\tilde{m}_{k+1}, \cdots, \tilde{m}_{k+1}$ can be successfully decoded provided that the error rate $\varepsilon_n$ satisfies $\varepsilon_n \lesssim 1/2$.

**(II) Improvement using permutation matrix $P_{K_r}$**

We assume here that the error vector in Section 4 is used. Then the following $2n \times 2n$ permutation matrix $P_{K_r}$ can be used for improving the security:

$$P_{K_r} = \begin{bmatrix} P_{\frac{n}{2}} & \mathbf{0} & P''_{\frac{n}{2}} \\ \mathbf{0} & I_n & \mathbf{0} \\ P'_{\frac{n}{2}} & \mathbf{0} & P'''_{\frac{n}{2}} \end{bmatrix}, \tag{47}$$

where $P_{\frac{n}{2}}$, $P'_{\frac{n}{2}}$, $P''_{\frac{n}{2}}$ and $P'''_{\frac{n}{2}}$ are random permutation matrices and $I_n$ is the $n \times n$ identity matrix. We also see that when the error vector $\mathbf{y}$ is given by Eq. (38) and error vectors $\mathbf{x}_L$ and $\mathbf{x}_R$ are given in a random manner, excluding $2r$ locations of the cyphertext $mMP_{K_2}$, then the error rate $\varepsilon_{2n}$ for $K_2$-PKC is given by

$$\varepsilon_{2n} = \frac{2n-4}{2n}. \tag{48}$$

As an example, for $K_2$-PKC, in order to decode $m_{k+1}$ and $m_{k+2}$ successfully, it is required, according to the choice of $\varepsilon = 2$ in Eq. (39), that the following relation holds for a random error vector $\mathbf{e}$ as shown below:

$$e_{\frac{n}{2}-1} = e_{\frac{n}{2}} = e_{\frac{3}{2}n-1} = e_{\frac{3}{2}n} = 0. \tag{49}$$

It should be noted that other components of $\mathbf{e}$ are chosen in a totally random manner.

**Theorem 2 :** If the error vectors for which $(\mathbf{e}_L, \mathbf{e}_R)$ is given by Eq. (38) are substituted by the randomized message vectors, then coding rate $\rho$ is given by

$$\begin{aligned} \rho &= \frac{k + r + \frac{n-\varepsilon}{3} + n - 2r}{2n} \\ &\cong (\frac{n}{3} + n + \frac{n}{2} - r)/2n \simeq \frac{11}{12}. \end{aligned} \tag{50}$$

□

**Remark 5 :** Using of $I_n$ in Eq. (47) implies that the public key $M$ is used without permutation, yielding a possibility of introducing a weakness to our proposed scheme here. We shall improve this problem in the following, (Ⅲ). □

**(Ⅲ) Improvement using permutation matrix $\tilde{P}_{K_r}$**

We assume here that the following permutation matrix $\tilde{P}_{K_r}$ is used:

$$\tilde{P}_{K_r} = \begin{bmatrix} P_{\frac{n}{2}} & \mathbf{0} & P''_{\frac{n}{2}} \\ \mathbf{0} & P_n & \mathbf{0} \\ P'_{\frac{n}{2}} & \mathbf{0} & P'''_{\frac{n}{2}} \end{bmatrix}, \tag{51}$$

where $P_n$ is an $n \times n$ permutation matrix. We also assume that the following generator polynomial is used for constructing the generator matrix $G$.

$$G(X) = G_{\mathrm{RS},D}(X)\Pi(X - \alpha_i), \tag{52}$$

where $G_{\mathrm{RS},D}(X)$ is the generator polynomial of Reed-Solomon (RS) code of the minimum distance $D$. We assume that the minimum distance $D$ satisfies the following:

$$D = 2t + 1 \tag{53}$$

In Eq. (52), we assume that $\alpha_i \in F_{2^m}$ is non-zero and randomly chosen

The generator polynomial $G(X)$ is very general in a sense that it generates the codes that include RS codes and totally random codes as particular cases.

**Example 2 :** $K$-matrix is given exactly in the same form of matrix as shown by Eq. (41) in Example 1. The $G_A$ and $\mathbf{r}_L$, $\mathbf{r}_R$, $\mathbf{r}'_L$, $\mathbf{r}'_R$ are given as follows:

$G_A$ **:** $256 \times 512$ matrix over $F_{2^8}$ that is generated with the polynomial $G(X)$ given by Eq. (52), where the minimum distance is given by $D = 201$.

$\mathbf{r}_L$, $\mathbf{r}_R$, $\mathbf{r}'_L$, $\mathbf{r}'_R$ **:** 256-symbols random vector over $F_{2^8}$.

The number of information symbols, $\kappa$, and the size of the cyphertext, $|C|$ are given by

$$\begin{aligned} &\kappa = (k+2) \times m = 2064 \ \text{(bits)} \\ &\text{and} \\ &|C| = 2n \times m = 8192 \ \text{(bits)}, \end{aligned} \tag{54}$$

respectively.

The size of the public-key, $S_{pk}$, is given by

$$S_{pk} = (k+2) \times 2n \times m = 2113536 \ \text{(bits)}. \tag{55}$$

The ambiguity of $G_A$ is given by

$$A_{G_A} = \log_2 \sharp\{G_A\}_C \simeq 8 \times 56 = 448 \ \text{(bits)}. \tag{56}$$

The ambiguity of $K$ matrix used for constructing $K_I$-PKC is given by

$$\begin{aligned} A_K &= \log_2 \sharp\{G_A\}_C + \log_2 \sharp\{(\mathbf{r}_L, \mathbf{r}_R)\}_C \\ &= 448 + 16384 = 16832 \ \text{(bits)}. \end{aligned} \tag{57}$$

In this case the probability that the randomly chosen $k = 258$ symbols of the part of the cyphertext with errors $\mathbf{y}_L$ and $\mathbf{y}_R$ have no error is given by $(\frac{412}{512})^{258} \simeq 4.4 \times 10^{-25}$, sufficiently small value.

**(Ⅳ) On Lee-Brickell attack**

When $h$ symbols are shown to be error free among the $2n$ symbols that constitutes the cyphertext over $F_{2^m}$, the message $m_1, m_2, \cdots, m_{k+r}$ can be estimated by searching all the possible patterns of the $k + r - h$ symbols. Let us now discuss on this matter in the following.

Let the probability that a randomly generated error symbols $\mathbf{e}_r$ assumes an element $\beta \in F_{2^m}$ be denoted by $P[\mathbf{e}_r = \beta]$. Obviously, this probability is given by

$$P[\mathbf{e}_r = \beta] = 2^{-m} \tag{58}$$

Consequently, the probability, $P_C(\varepsilon)$, that the obtaining of a error vector correctly is given by

$$P_C(\varepsilon) = (2^{-m})^{(k+r-h)}. \tag{59}$$

Let us consider the Lee-Brickell Attack [4] on our $K_2$-PKC given in Example 1. Evidently each of the remaining 125 symbols has an error with probability $1 - 2^{-7}$. In order to perform Lee-Brickell attack, we have to estimate $129 - 4 = 125$ error free symbols in

$\mathbf{y}_L$ and $\mathbf{y}_R$ in an exhaustive manner. The probability of the obtaining of error symbols is given by $2^{-125\cdot 7} = 2^{-875}$, an extremely small value.

In Example 2, while the error rate in $(\mathbf{x}_L, \mathbf{x}_R)$ is approximately 1.0, the error rate in $(\mathbf{y}_L, \mathbf{y}_R)$ is only $\frac{100}{512} = 0.195$. Thus the corresponding part of the cyphertext may be threatened by Lee-Brickell attack. However the probability of the obtaining of error free symbols among 512 symbols is given by $\binom{412}{258}/\binom{512}{258} = 1.59 \times 10^{-36}$, sufficiently small value.

**(V) Improvements by letting error rate be smaller** $K_r$-PKC given in Example 1, we have chosen the error rate $\rho$ in $\mathbf{x}_L$ and $\mathbf{x}_R$ to be approximately 1.0. Consequently the locations of $2r$ error free symbols should be made public. However the publication of error free symbols may introduce a certain weakness in our $K_r$-PKC. To overcome this problem, the simplest way is to make the error rate smaller. We shall discuss on this matter using the numerical example for $K_r$-PKC given in Example 2.

We let error rate in $\mathbf{x}_L$ and $\mathbf{x}_R$ be $\frac{3}{4}$, yielding error free rate in $(\mathbf{x}_L + \mathbf{y}_R; \mathbf{x}_R + \mathbf{y}_L)$ to be $\frac{1}{4} \cdot \frac{156}{256} = 0.152$.

The expectation of the number of error free symbols in the above vector is given by 67.8. The standard variation is given by $\sqrt{512 \times 0.152 \times 0.848} = 8.124$. The probability of obtaining less than 8 error free symbols, and more than 129 error free symbols $P[x \leq 6, x \geq 128]$, is then given by $1.54 \times 10^{-13}$, sufficiently small value.

When error symbols are substituted by message symbols, the coding rate $\rho$ can be given as

$$\rho = \frac{n + r + \frac{1}{m}\log_2\binom{n}{t} + t + \frac{1}{m}\log_2\binom{n}{n\varepsilon_n} + n\varepsilon_n}{2n}. \quad (60)$$

For example, the coding rate in Example 2 is now given by $\rho = 0.82$, sufficiently large value.

### 5.2 Computational Efforts on Encryption and Decryption

In this section, we briefly discuss on the computational efforts on Encryption and Decryption for the proposed $K_r$-PKC. First, we shall define the following symbol:

$R_g$ **:** Residue class ring modulo a polynomial $G(X)$ of degree $g$ over $F_{2^m}$.

**a(X),b(X),c(X) :** Elements of $R_g$.

In the following, we assume that $r = 1$. A generalization to the case where $r > 1$ is straightforward. For $K_r$-PKC with error vector given by (38), basically, the following operations are performed.

**For encryption:**

**(1)** $\mathbf{m}M$, yielding $\mathbf{m}M$.

**(2)** $\mathbf{e} + \mathbf{m}M$, yielding $\mathbf{C}$ [Eq. (9)].

**For decryption:**

**(1)** $(\mathbf{e} + mM)P^{-1}$.

**(2)** $(\tilde{m}K)_L + \mathbf{e}_L + (\tilde{m}K)_R + \mathbf{e}_R$.

**(3)** Assuming $\hat{\tilde{m}}_{k+1}$ in $2^m$ ways on $\tilde{m}_{k+1}(\mathbf{r}_L, \mathbf{r}_R) + \mathbf{C}P^{-1}$, yielding $\tilde{m}_{k+1}$ and $\mathbf{x}_L + \mathbf{y}_R$..

**(4)** $\mathbf{F}_G + \mathbf{x}_L + (\mathbf{x}_L + \mathbf{y}_R, \mathbf{0})$, yielding $\mathbf{F}_G + (\mathbf{y}_R, \mathbf{y}_L)$.

**(5)** $\{\mathbf{F}_G(X) + y(X)^3\}^{3^{-1}} \equiv y(X) \bmod G(X)$, yielding $(\mathbf{y}_R, \mathbf{y}_R)$.

**(6)** $\mathbf{x}_L + \mathbf{y}_R + \mathbf{y}_R = \mathbf{x}_L$.

**(7)** $x_L(X)X^g \cdot X^{-g} \equiv x_L(X) \bmod G(X)$.

**(8)** $CP^{-1} + (\mathbf{x}_L, \mathbf{y}_L, 0, 0)$, yielding $\tilde{m}G_A$.

**(9)** $(\tilde{m}G_A)G_A^{-1}$, yielding $\tilde{m}$.

**(10)** $\tilde{m}S$, yielding $m$.

## 6 Concluding Remarks

We have proposed a new class of PKC using algebraic coding on the basis of superimposition and randomness. The proposed schemes have the following features:

The freedom of choosing $K$-matrix is larger than that of the choosing of generator matrix based on the algebraic error correcting codes such as Goppa codes [2], under the same size of public-key.

It seems that our proposed PKC, $K_r$-PKC with extremely large number of errors has improved the security level significantly compared with the conventional scheme. For example, in the proposed PKC, $K_r$-PKC probability of the estimating of error free $k+r$ message successfully can be made sufficiently small. Besides the proposed scheme has opened up a new problem of constructing a new class of PKC based on algebraic coding with high coding rate.

## A Appendix: Simple $K_r$-PSK

In this appendix, we shall present the simplest $K_r$-PKC where $(k+r) \times n$ $K$ matrix is given as follows:

$$K_S = \begin{bmatrix} G & \mathbf{0} \\ R & \end{bmatrix} \quad (61)$$

In Eq. (61), we assume that $G$ is generated by a generator polynomial $G(X)$. The $\mathbf{0}$ is $r \times 2n$ zero matrix and $R$ is an $r \times (2n + r)$ random matrix. It should be noted that the matrix $\mathbf{0}$ can be located any place.

However this simple structure of $K$-matrix, $K_S$, may cause another problem of introducing a certain weakness in our scheme unless message symbols are sufficiently randomized. Further investigation has been left for future.

## References


[1] R.J. McEliece: "A public-key cryptosystem based on algebraic coding theory", DSN Progress Report 42–44, pp 114–116, Jet Propulsion Lab., Pasadena, CA, (1978).

[2] F.J. MacWilliams, N.J.A. Sloane: "The Theory of Error-Correcting Codes", North-Holland, Amsterdam, (1997).

[3] H. Niderreiter: "Error-correcting codes and cryptography", in "Public-Key Cryptography and Computational Number Theory" edited by K. Alster, J. Urbanowicz and H.C. Williams, Walter de Gruyter, (2001).

[4] P.J. Lee, E.F. Brickell: "An observation on the security of McEliece's public-key cryptosystem", in "Advances in Cryptology – EUROCRYPT '88" edited by C.G. Günther, Lecture Notes in Comput. Sci. 330, 275–280, Springer, Berlin, (1998).

[5] K. Kobara, H. Imai: "Semantically Secure Public-Key Cryptosystems Based on the Decoding Problem of Linear Codes", IEICE Trans. on Fundamentals, Vol. J87-A, No. 7, pp. 870–880, (2004).

[6] M. Kasahara: "A Construction of Public-Key Cryptosystem Based on Algebraic Coding and Random Coding", proceeding of 27th SITA, pp. 119–122, (2004).

[7] M. Kasahara: "A Construction of Public-Key Cryptosystem Based on Algebraic Coding and Random Coding Techniques over $F_{2^m}$", ISEC 2004-81, pp. 21–26, (2004).