A Construction of 100 bit Public-Key Cryptosystem and Digital Signature Scheme

Masao KASAHARA * Ryuichi SAKAI †

March 10, 2003

Abstract

Extensive studies have been made of the public-key cryptosystems based on multivariate polynomials . However most of the proposed public-key cryptosystems of rate 1.0 based on multivariate polynomials, are proved not secure. In this paper, we propose several types of new constructions of public-key cryptosystems based on two classes of randomly generated simultaneous equations, namely, a class based on bijective transformation and another class based on random transformation. One of the features of the proposed cryptosystems is that the sets of random simultaneous equations significantly improve the utilization factor of the public-key space. We show an example of the proposed cryptosystem whose size is only 100 bits that seems to be apparently secure in a sense that the utilization factor is significantly large compared with the conventional public-key cryptosystems.

1 Introduction

Extensive studies have been made of the Public-Key Cryptosystem(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 whose security seems to depend on the difficulty of the problem of Solving Simultaneous Equations(SSE) of degree larger than or equal to 2. Hereinafter simultaneous equation will be abbreviated by SE. Accordingly we shall refer to the conventional scheme, constructed based on Simultaneous Equations(SE) of degree g will be referred to as SE(g)-PKC. The simultaneous equations used in the proposed PKC are generated in a random manner, in a sharp contrast with the conventional methods whose security also seem to be related to the difficulty of SSE[1][2]. Because our proposed PKCs are constructed, based on the Random Simultaneous Equations(RSE) of degree g, we shall refer to our proposed scheme as RSE(g)-PKC, for short.

It should be noted that all the proposed PKC in this paper has no redundancy, i.e., the message transmission rate of any proposed PKC in this paper is completely 100%. Thus our proposed schemes will be mainly compared with PKC with no redundancy.

2 Construction of RSE(g)-PKC

Letting message vector over $\mathbb{F}_q$ be denoted by

$$\boldsymbol{x} = (x_1, x_2, \cdots, x_n), \tag{1}$$

where $x_i \in \mathbb{F}_q \quad (i = 1, 2, \cdots, n)$.

The following relation implies that the message vector $\boldsymbol{x}$ takes on a certain message vector $\tilde{\boldsymbol{x}}$:

$$\boldsymbol{x} = \tilde{\boldsymbol{x}} = (\tilde{x_1}, \tilde{x_2}, \cdots, \tilde{x_n}) \tag{2}$$

where we assume that the variable $x_i \in \mathbb{F}_q$ takes on the actual value $\tilde{x}_i$. In the followings, symbols $\tilde{y}_i$ and $\tilde{z}_i$ will be used in an exactly similar manner as $\tilde{x}_i$.

One of the simplest version of the RSE(g)-PKC can be constructed in the following manner where we let $q = 2$ and $g = 2$:

*Osaka Gakuin University, Kishibe Osaka,Japan

†Osaka Electro-Communication University, 18-8 Hatsucho Neyagawa-shi Osaka Japan

Step 1 : Letting the message vector over $\mathbb{F}_2$ be given by $\boldsymbol{x} = (x_1, x_2, \cdots, x_n)$, the $\boldsymbol{x}$ is transformed to vector $\boldsymbol{y}$ as follows:

$$\boldsymbol{x}A = \boldsymbol{y}, \tag{3}$$

where A is an $n \times n$ non-singular matrix over $\mathbb{F}_2$ and $\boldsymbol{y}$ will be denoted by

$$\boldsymbol{y} = (y_1, y_2, \cdots, y_n), \tag{4}$$

where $y_i \in \mathbb{F}_2$.

Step 2 : The components of the vector y are partitioned into N sub vectors, yielding the following vector :

$$\boldsymbol{y} = (\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_N), \tag{5}$$

where $\boldsymbol{Y}_j$ is given by

$$\boldsymbol{Y}_j = (y_{j1}, y_{j2}, \cdots, y_{jt}). \tag{6}$$

Definition 1 : The following transformation:

$$\phi(\boldsymbol{u}) = \boldsymbol{v} \tag{7}$$

is referred to as "non-singular", if and only if the transformation has the following inverse-transformation:

$$\phi^{-1}(\boldsymbol{v}) = \boldsymbol{u}. \tag{8}$$

$\square$

Evidently, the transformation ϕ is bijective.

Step 3 : Given $\boldsymbol{Y}_j$ $(j = 1, 2, \cdots, N)$, the following transformation, $\phi(\boldsymbol{y}) = \boldsymbol{z}$, is performed on the basis of "randomness" :

$$\begin{aligned}
z_{j1} &= R^{(2)}_{j1}(y_{j1}, y_{j2}, \cdots, y_{jt}),\\
z_{j2} &= R^{(2)}_{j2}(y_{j1}, y_{j2}, \cdots, y_{jt}),\\
&\vdots\\
z_{ji} &= R^{(2)}_{ji}(y_{j1}, y_{j2}, \cdots, y_{jt}),\\
&\vdots\\
z_{jt} &= R^{(2)}_{jt}(y_{j1}, y_{j2}, \cdots, y_{jt}),
\end{aligned} \tag{9}$$

where $z_{ji} = R^{(2)}_i(y_{j1}, y_{j2}, \cdots, y_{jt})$ is a quadratic equation in t variables, $y_{j1}, y_{j2}, \cdots, y_{jt}$. We assume that the coefficients of the equation is chosen in a random manner under the condition that the above transformation, $\phi(\boldsymbol{y})$, is non-singular. In such case where SE is required to be non-singular, we shall refer to SE as first class SE. Evidently Eq.(9) proves to be Simultaneous Equation(SE) of degree 2 in t variables.

Step 4 : Given $\boldsymbol{Y}_1$, another transformation, not necessarily bijective, is performed, yielding the following SE of degree 2 :

$$\begin{aligned}
w_{11} &= r^{(2)}_{11}(y_{11}, y_{12}, \cdots, y_{1t}),\\
w_{12} &= r^{(2)}_{12}(y_{11}, y_{12}, \cdots, y_{1t}),\\
&\vdots\\
w_{1i} &= r^{(2)}_{1i}(y_{11}, y_{12}, \cdots, y_{1t}),\\
&\vdots\\
w_{1t} &= r^{(2)}_{1t}(y_{11}, y_{12}, \cdots, y_{1t}).
\end{aligned} \tag{10}$$

We assume here that the coefficients of the above SE given by Eq.(10) are generated in a totally random manner in a sense that SE do not have to be non-singular. In such case we shall refer to SE as second class SE.

Step 5 : Given $\boldsymbol{Y}_1, \boldsymbol{Y}_2, \cdots, \boldsymbol{Y}_j$, the following second class SE are generated in a similar manner as in Step 4:

$$\begin{aligned}
w_{j1} &= r^{(2)}_{j1}(y_{11}, \cdots, y_{1t}, y_{21}, \cdots, y_{2t}, \cdots, y_{j1}, \cdots, y_{jt}),\\
w_{j2} &= r^{(2)}_{j2}(y_{11}, \cdots, y_{1t}, y_{21}, \cdots, y_{2t}, \cdots, y_{j1}, \cdots, y_{jt}),\\
&\vdots\\
w_{ji} &= r^{(2)}_{ji}(y_{11}, \cdots, y_{1t}, y_{21}, \cdots, y_{2t}, \cdots, y_{j1}, \cdots, y_{jt}),\\
&\vdots\\
w_{jt} &= r^{(2)}_{jt}(y_{11}, \cdots, y_{1t}, y_{21}, \cdots, y_{2t}, \cdots, y_{j1}, \cdots, y_{jt}).
\end{aligned} \tag{11}$$

We also assume that the above SE(second class) are random in a sense that the coefficients are chosen in a random manner.

Step 6 : From $w_{j1}, w_{j2}, \cdots, w_{jt}$, the followings are generated for $j = 0, 1, \cdots, N$:

$$
\begin{aligned}
\gamma_{j1} &= z_{j1} + w_{(j-1)1}, \\
\gamma_{j2} &= z_{j2} + w_{(j-1)2}, \\
&\vdots \\
\gamma_{ji} &= z_{ji} + w_{(j-1)i}, \\
&\vdots \\
\gamma_{jt} &= z_{jt} + w_{(j-1)t}.
\end{aligned} \tag{12}
$$

where $w_{01}, w_{02}, \cdots, w_{0t}$ are all zeros.

Step 7 : Letting $\mathbf{\Gamma} = (\mathbf{\Gamma}_1, \mathbf{\Gamma}_2, \cdots, \mathbf{\Gamma}_N)$ where $\mathbf{\Gamma}_j = (\gamma_{j1}, \gamma_{j2}, \cdots, \gamma_{jt})$, the following final transformation is performed :

$$
\mathbf{\Gamma} B = (K_1, K_2, \cdots, K_n), \tag{13}
$$

yielding the set of public-keys, $K = (K_1, K_2, \cdots, K_n)$, where B is an $n \times n$ non-singular matrix over $\mathbb{F}_2$.

The public-keys $\{K_i\}$ can be denoted as :

$$
\begin{aligned}
K_1 &= f_1^{(2)}(x_1, x_2, \cdots, x_n), \\
K_2 &= f_2^{(2)}(x_1, x_2, \cdots, x_n), \\
&\vdots \\
K_i &= f_i^{(2)}(x_1, x_2, \cdots, x_n), \\
&\vdots \\
K_n &= f_n^{(2)}(x_1, x_2, \cdots, x_n),
\end{aligned} \tag{14}
$$

where $f_i^{(2)}(x_1, x_2, \cdots, x_n)$ is a quadratic equation obtained through Steps 1 to 7.

3 Encryption and Decryption

3.1 Encryption

Encryption can be performed simply by substituting $\tilde{x}_i$ for x_i in Eq.(14), yielding the cipher-text $\boldsymbol{c} = (c_1, c_2, \cdots, c_n)$ over $\mathbb{F}_2$. In Fig.1, we show an example of an encoder of RSE(2)-PKC where we assume that $t = 3$, $N = 3$, $n = 9$.

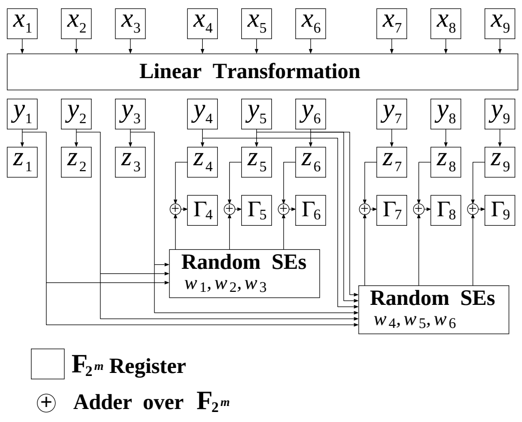

Fig. 1: An example of the principal part of the encoder of RSE(2)-PKC

3.2 Decryption

Decryption can be performed through the following Steps.

Step 1 : Cipher-text $\boldsymbol{c}$ is transformed to $\tilde{\boldsymbol{\Gamma}}$ as follows:

$$\boldsymbol{c}B^{-1} = \tilde{\boldsymbol{\Gamma}} \tag{15}$$

where $\tilde{\boldsymbol{\Gamma}}$ is partitioned into N sub-vectors each of which has same dimension of t.

$$\tilde{\boldsymbol{\Gamma}} = (\tilde{\boldsymbol{\Gamma}}_1, \tilde{\boldsymbol{\Gamma}}_2, \cdots, \tilde{\boldsymbol{\Gamma}}_N) \tag{16}$$

Step 2 : The $\tilde{\boldsymbol{\Gamma}}_1$ is evidently $(\tilde{z}_{11}, \tilde{z}_{12}, \cdots, \tilde{z}_{1t})$. Thus $(\tilde{z}_{11}, \tilde{z}_{12}, \cdots, \tilde{z}_{1t})$. is inverse-transformed to $\tilde{\boldsymbol{Y}}_1 = (\tilde{y}_{11}, \tilde{y}_{12}, \cdots, \tilde{y}_{1t})$ by a table look-up method. As the table look-up method is used in the decoding process, from the practical point of view, t is recommended to satisfy

$$t \lesssim 10. \tag{17}$$

Step 3 : Given $\tilde{\boldsymbol{Y}}_1 = (\tilde{y}_{11}, \tilde{y}_{12}, \cdots, \tilde{y}_{1t})$, the transformation of Eq(10) is performed on $\tilde{\boldsymbol{Y}}_1$, yielding $\tilde{\boldsymbol{W}}_1 = (\tilde{w}_{11}, \tilde{w}_{12}, \cdots, \tilde{w}_{1t})$.

Step 4 : Given $\tilde{\boldsymbol{W}}_1$, the vector $\tilde{\boldsymbol{Z}}_2 = (\tilde{z}_{21}, \tilde{z}_{22}, \cdots, \tilde{z}_{2t})$ is decoded as follows :

$$\tilde{\boldsymbol{Z}}_2 = \tilde{\boldsymbol{W}}_1 + \tilde{\boldsymbol{\Gamma}}_2 \tag{18}$$

Step 5 : The vector $\tilde{\boldsymbol{Z}}_2$ is inverse-transformed to $\tilde{\boldsymbol{Y}}_2 = (\tilde{y}_{21}, \tilde{y}_{22}, \cdots, \tilde{y}_{2t})$.

$i \leftarrow 2$.

Step 6 : From $(\tilde{\boldsymbol{Y}}_1, \cdots, \tilde{\boldsymbol{Y}}_i)$, $\tilde{\boldsymbol{W}}_i = (\tilde{w}_{i1}, \tilde{w}_{i2}, \cdots, \tilde{w}_{it})$ is decoded. From $\tilde{\boldsymbol{W}}_i$ and $\tilde{\boldsymbol{\Gamma}}_{i+1}$, $\tilde{\boldsymbol{Z}}_{i+1}$ is obtained. From $\tilde{\boldsymbol{Z}}_{i+1}$, $\tilde{y}_{i+1}$ is finally decoded.
$i \leftarrow i + 1$

Step 7 : While $i < N$, repeat Step 6

Step 8 : Through Steps 6 to 7, $\tilde{\boldsymbol{Y}} = (\tilde{\boldsymbol{Y}}_1, \tilde{\boldsymbol{Y}}_2, \cdots, \tilde{\boldsymbol{Y}}_N)$ is decoded. The vector $\tilde{X}$ is decoded by the inverse-transformation of Eq.(3), yielding a message vector as follows :

$$\tilde{\boldsymbol{y}}A^{-1} = \tilde{\boldsymbol{x}} = (\tilde{x}_1, \tilde{x}_2, \cdots, \tilde{x}_n). \tag{19}$$

4 Efficiency of Public-Key Space

4.1 Size of Public-Key

Let us suppose that the transformation , $\boldsymbol{y} \to \boldsymbol{z}$, can be performed as follows [1],[2]:

$$\boldsymbol{Y}_i^{2^h} \cdot \boldsymbol{Y}_i^{2^l} = \boldsymbol{Z}_i, \tag{20}$$

where we assume that $\boldsymbol{Y}_i \in \mathbb{F}_{2^t}$ and $0 \leq h < l \leq t-1$ holds.

The set of t components of $\boldsymbol{Z}_i$, obviously, constitutes the non-singular SE(2)s.

Theorem 1 : The size of the public-key of SE(2)-PKC, S_{PK}, is given by

$$S_{PK} = n({}_nH_2 + 1) = n({}_{n+1}C_2 + 1), \tag{21}$$

where n is the block-length of the cipher-text. □

For example, when $n = 100$, the size of S_{PK} is given by 505,000 bits or 505 K bits.

4.2 Effective Size of Public-Key

Theorem 2 : The total number of the possible transformation of Eq.(20) is given by

$$n_{syst}(t) = \frac{1}{t}\varphi(2^t - 1)_t C_2, \tag{22}$$

where $\varphi(x)$ is the Euler's function of x. □

Proof : The total number of the possible different primitive polynomials that generate different $\mathbb{F}_{2^t}$s is given by $\frac{1}{t}\varphi(2^t - 1)$. The exponent $2^h + 2^l$ in Eq.(20), assumes ${}_tC_2$ different values, yielding the proof. □

Corollary to Theorem 2 : The total number of the possible different set of public-key, N_{PK} , for SE(2)-PKC is given by

$$N_{PK} = \left\{ \frac{1}{t} \varphi(2^t - 1)_t C_2 \right\}^N \tag{23}$$

□

Definition 2 : Effective size of the public-key based only on the transformation of Eq.(20), $\tilde{S}_{PK}(\text{effec.})$ is defined as

$$\tilde{S}_{PK}(\text{effec.}) = \log_2 N_{PK} \qquad \text{(in bits)} \tag{24}$$

□

For example, when $n = 100$, $t = 4$, the effective size of the public-key of the SE(2)-PKC is given by

$$\begin{aligned} \tilde{S}_{PK}(\text{effec.}) &= \log_2 \left\{ \frac{1}{4} \varphi(2^4 - 1)_4 C_2 \right\}^{25} \\ &\fallingdotseq 89.6 \text{ (in bits)}. \end{aligned} \tag{25}$$

As an another example, when $n = 100$, $t = 100$, $\tilde{S}_{PK}(\text{effec.})$ is

$$\begin{aligned} \tilde{S}_{PK}(\text{effec.}) &= \log_2 \left\{ \frac{1}{100} \varphi(2^{100} - 1)_{100} C_2 \right\} \\ &\fallingdotseq 104.5 \text{ (in bits)}. \end{aligned} \tag{26}$$

We see that the effective size of the public-key of the conventional public-key SE(2)-PKC takes on a very small value, although the size of the public-key space(in bits) is given by 505000 bits or 505K bits.

We shall show, in the following, that our proposed RE(2)-PKC yields a very large effective size of public-key, in a sharp contrast with the conventional SE(2)-PKC.

Let the set of non-singular SE(2)s be denoted by $\{\text{SE}(2)\}$ and the order of $\{\text{SE}(2)\}$, by $\#\{\text{SE}(2)\}$. When the order of $\{y_i\}$, t, is 4, $\#\{\text{SE}(2)\}$ is given by [3],

$$\#\{\text{SE}(2)\} = 2016840. \tag{27}$$

Besides the SE(2)s mentioned above, another class of SE(2)s, second class SEs that are not necessarily non-singular are used in RSE(2)-PKC, for generating a set of public-keys. It should be noted that the using of the second class of SEs significantly improves the effective size of public key.

Let the total number of different random SE(2)s in $t \cdot i$ variables be denoted by $n_{\text{random}}(t \cdot i)$. The total number of random SE(2)s used in RSE(2)-PKC for t and n, $N_{\text{random}}(n, t)$ is given by

$$N_{\text{random}}(n, t) = \prod_{i=1}^{N-1} n_{\text{random}}(t \cdot i). \tag{28}$$

For example, when $n = 100$, $t = 4$, the total number of random SE(2)s used in RSE(2)-PKC is

$$N_{\text{random}}(100, t) = \prod_{i=1}^{24} n_{\text{random}}(4 \cdot i), \tag{29}$$

where $n_{\text{random}}(4 \cdot i)$ is given by

$$n_{\text{random}}(4 \cdot i) = 2^{({}_{4i}H_2 + 1)4}. \tag{30}$$

For example, when $i = 1$, the vector $(w_{11},\ w_{12},\ w_{13}, w_{14})$ in Eq.(10) is the $({}_4H_2 + 1)4$ dimensional vector over $\mathbb{F}_2$. Thus $\{(w_{11}, w_{12}, w_{13}, w_{14})\}$ is the set of $2^{({}_4H_2+1)4}$ different vectors over $\mathbb{F}_2$.

Definition 3 : The effective size of the public-key of RSE(2)-PKC, for n and t, is given by

$$S_{PK}(\text{effec.}) = \tilde{S}_{PK}(\text{effec.}) + \log_2 N_{\text{random}}(n, t). \tag{31}$$

□

For example, for $n = 100$, $t = 4$, $S_{PK}(\text{effec.})$ is given by

$$\begin{aligned} S_{PK}(\text{effec.}) &= 523.6 + \sum_{i=4}^{24} ({}_{4i}H_2 + 1)4 \\ &\fallingdotseq 159820 \text{ in bits} \end{aligned} \tag{32}$$

We see that the proposed RSE(2)-PKC of length 100 realizes a very large effective size of public-key.

Definition 4 : Utilization factor of the public-key space is defined as

$$\eta(PK_{\text{space}}) = \frac{S_{PK}(\text{effec.})}{S_{PK}} \tag{33}$$

□

For example, for $n = 100$ and $t = 4$, the utilization factor of the public-key space of the proposed RSE(2)-PKC is

$$\eta(PK_{\text{space}}) = \frac{158878}{505000} \fallingdotseq 0.316. \tag{34}$$

On the other hand, for the same value of n and t, the utilization factor of the public-key space of SE(2)-PKC achieves only

$$\eta(PK_{\text{space}}) \fallingdotseq \frac{98.6}{505000} \fallingdotseq 0.000177, \tag{35}$$

yielding a very small value compared with the proposed scheme.

For $n = 100$ and $t = 100$, the $S_{PK}(\text{effec.}) = \tilde{S}_{PK}(\text{effec.})$ for the conventional SE(2)-PKC, is given by 104.5 bits. Thus even in this case, $\eta(PK_{\text{space}})$ is

$$\eta(PK_{\text{space}}) \fallingdotseq \frac{104.5}{505000} \fallingdotseq 0.000207. \tag{36}$$

In Fig.2, we show the effective size of public-keys, $S_{PK}(\text{effec.})$ for several classes of SE(2)-PKC and RSE(2)-PKC. Semi-RSE(2)-PKC is the RSE(2)-PKC where only the first class of transformation, (9), is used.

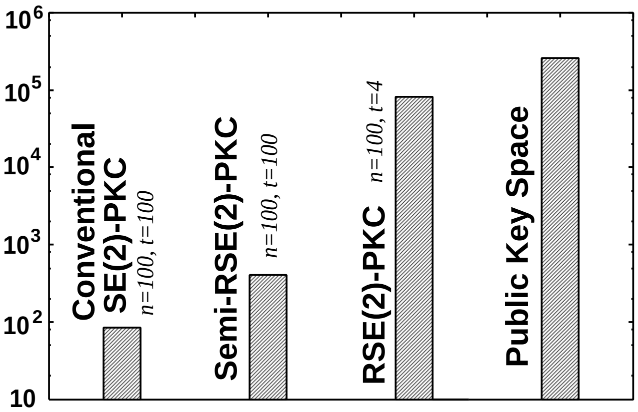

Fig. 2: Effective size of public-key for SE(2)-PKC and RSE(2)-PKC

5 Construction of RSE(g)-PKC over $\mathbb{F}_{2^m}$

We also present here a construction of RSE(g)-PKC over $\mathbb{F}_{2^t}$ ($m \geq 1$) [5]. We shall show in the following two examples.

Remark : The number of variables in Example 1 is 6. Thus the PKC in Example 1 cannot be secure enough as it might be broken with the attack based on Gröbner basis. It should be also noted that in these examples first class SE has not be found yet, thus we shall show only the lower bound of $S_{PK}(\text{effec.})$.

Example 1 : RSE(5)-PKC over $\mathbb{F}_{2^{17}}$ where $n = 6$ and $t = 2$.

The size of the public-key of RSE(5)-PKC, S_{PK}, is given by

$$S_{PK} = 17n \sum_{i=0}^{5} {}_6H_i = 102 \sum_{i=0}^{5} {}_6H_i = 47124 \quad \text{(bits)}. \tag{37}$$

We see that the size of the public-key becomes very small compared with the S_{PK}, 535908 (bits) for the RSE(2)-PKC over $\mathbb{F}_2$ where $n = 102$. For this example, the exact value of $\tilde{S}_{PK}(\text{effec.})$ has not beren found yet, at the moment, we shall use here the lower bound of $S_{PK}(effec.)$. The lower bound of the effective size of the public-key, $\underline{S_{PK}}$(effec.) in this case is given by

$$\underline{S_{PK}}(\text{effec.}) > 17 \cdot 2 \sum_{i=0}^{5} ({}_2H_i + {}_4H_i) = 4998 \quad \text{(bits)}. \tag{38}$$

The lower bound of the utilization factor $\eta(PK_{\text{space}})$ is given by

$$\eta(PK_{\text{space}}) \fallingdotseq 0.106. \tag{39}$$

Example 2 : RSE(5)-PKC over $\mathbb{F}_{2^{13}}$ where $n = 10$, $t = 2$.
The size of the public-key of RSE(5)-PKC, S_{PK}, is given by

$$S_{PK} = 13n \sum_{i=0}^{5} {}_{10}H_i = 130 \sum_{i=0}^{5} {}_{10}H_i = 390390 \quad \text{(bits)}. \tag{40}$$

We see that the size of the public-key becomes shorter compared with the S_{PK}, 1107080 (bits) for the RSE(2)-PKC over $\mathbb{F}_2$ where $n = 130$.
The lower bound of the effective size of the public-key, $\underline{S_{PK}}$(effec.) in this case is given by

$$\begin{aligned} \underline{S_{PK}}(\text{effec.}) &> 13 \cdot 2 \sum_{i=0}^{5} ({}_2H_i + {}_4H_i + {}_6H_i + {}_8H_i) \\ &= 49296. \end{aligned} \tag{41}$$

The lower bound of the utilization factor $\eta(PK_{\text{space}})$ is given by

$$\eta(PK_{\text{space}}) \fallingdotseq 0.126. \tag{42}$$

6 Open Problem

We shall upload the open problems in WWW site "http://www.osaka-gu.ac.jp/php/kasahara/publickey.htm". The parameters are set as follows:

$$\begin{aligned} n &= 100 \\ t &= 4 \text{ or } 5 \\ N &= 25 \text{ or } 20 \end{aligned}$$

The public-keys $K_0, K_1, \cdots, K_{99}$ are represented by the following orders.

$$K_l = (k_{0,0}^{(l)} k_{0,1}^{(l)} \cdots k_{0,99}^{(l)} \cdots k_{i,i}^{(l)} k_{i,i+1}^{(l)} \cdots k_{i,99}^{(l)} \cdots k_{98,99}^{(l)} k_{99,99}^{(l)})_2$$

where $k_{i,j}^{(l)}$ is a coefficient of $x_i x_j$ of the public-key K_l.

7 Implementation

In this chapter, we shall discuss on the software implementation of the proposed RSE(g)-PKC. We shall use the word-based bit by bit exclusive OR operations for the software implementation, using the 32 bits word processor.

When the parameter n is equal to 80, the total costs of the computation of our implementation(see the Appendix) of the encryption is on an average $2 \times 80 \times 80/2 = 80^2 = 6400$ times word-based bit by bit EXOR operations, under the assumption that the average value of the Hamming weight of the message is equal to 40. [1]

We have examined the performance of the enciphering of our public-key cryptosystem, RSE(2)-PKC, under the following environments.

Table 1 Computation environment

CPU	Pentium III or Pentium IV
OS	Linux
Compiler	gcc-3.2

Table 2 Computation time for encryption

Block size : n	80	100
Pentium III 700MHz	0.036ms	0.08ms
Pentium IV 2.4GHz	0.005ms	0.009ms

In case of Pentium3, each EXOR needs 3.9 clocks and in case of Pentium4, each EXOR needs 1.9 clocks.

[1] It should be noted that the number of word-based bit by bit EXOR operations for the encryption can be reduced to $4080 \fallingdotseq {}_{80}H_2 \cdot 80/32/2$(see the Appendix).

8 Conclusion

We have presented a new class of public-key cryptosystem referred to as RSE(g)-PKC. Although the details of doing so are omitted, we can show that the digital signature scheme can be easily realized with our proposed PKC.

In this paper, we have discussed primarily on RSE(2)-PKC over $\mathbb{F}_2$. However the proposed RSE(g)-PKC can be generalized in various ways. For example, we can construct RSE(g)-PKC over $\mathbb{F}_{2^m}$ or $\mathbb{F}_{q^m}$. In order to speed-up the encoding and decoding procedure, semi-RSE(g)-PKC could be used, under the condition that the effective size of public-keys is sufficiently large. Various interesting studies have been left for the future.

References

[1] N. Koblitz : "Algebraic Aspects of Cryptography", Springer-Verlag Berlin Heidelberg,(1998).

[2] T. Matsumoto and H. Imai : "Public quadratic Polynomial-tuples for efficient signature-verification and message-encryption", Proc. of Eurocrypt '88, Springer-Verlag, 419-453, (1989).

[3] M. Kasahara and R. Sakai : "Notes on public key cryptosystem based on multivariate polynomials of high degree", Technical Report of IEICE, ISEC 2001-64 (2001-9).

[4] M. Kasahara and R. Sakai : "A construction of a new public key cryptosystems on the basis of multivariate polynomials of high degree – A method for yielding short public key cryptosystem and short digital signature scheme –", Technical Report of IEICE, ISEC 2002-67 (2002-9).

[5] M. Kasahara and R. Sakai : " A Construction of Short Public-Key Cryptosystem over Extension Field ", Technical Report of IEICE, ISEC 2003-(to-appear) (2003-3).

Appendix

The public key is represented as an n quadratic equations each of which has n variables. We shall denote the coefficient of the term $x_i x_j$ by $k_{i,j}$ where $i \leq j$.

In the following, we shall explain the example of $n = 8$ for simplicity. For the fast computation of the software implementation, the i-th quadratic equations of the public key, K_i, is divided into 8 parts as follows:

$$K_i = K_{i0} + K_{i1} + K_{i2} + K_{i3} + K_{i4} + K_{i5} + K_{i6} + K_{i7},$$

$$\begin{aligned}
K_{i0} &= x_0(k_{0,0}x_0 + k_{0,1}x_1 + k_{0,2}x_2 + k_{0,3}x_3 + k_{0,4}x_4), \\
K_{i1} &= x_1(k_{1,1}x_1 + k_{1,2}x_2 + k_{1,3}x_3 + k_{1,4}x_4 + k_{1,5}x_5), \\
K_{i2} &= x_2(k_{2,2}x_2 + k_{2,3}x_3 + k_{2,4}x_4 + k_{2,5}x_5 + k_{2,6}x_6), \\
K_{i3} &= x_3(k_{3,3}x_3 + k_{3,4}x_4 + k_{3,5}x_5 + k_{3,6}x_6 + k_{3,7}x_7), \\
K_4 &= x_4(k_{4,4}x_4 + k_{4,5}x_5 + k_{4,6}x_6 + k_{4,7}x_7), \\
K_{i5} &= x_5(k_{5,5}x_5 + k_{5,6}x_6 + k_{5,7}x_7 + k_{0,5}x_0), \\
K_{i6} &= x_6(k_{6,6}x_6 + k_{6,7}x_7 + k_{0,6}x_6 + k_{1,6}x_1), \\
K_{i7} &= x_7(k_{7,7}x_7 + k_{0,7}x_9 + k_{1,7}x_1 + k_{2,7}x_2).
\end{aligned}$$

The binary representation of the above 8 equations is given by

$$\begin{aligned}
K[i][0] &= (k_{0,0}k_{0,1}k_{0,2}k_{0,3}k_{0,4}), \\
K[i][1] &= (k_{1,1}k_{1,2}k_{1,3}k_{1,4}k_{1,5}), \\
K[i][2] &= (k_{2,2}k_{2,3}k_{2,4}k_{2,5}k_{2,6}), \\
K[i][3] &= (k_{3,3}k_{3,4}k_{3,5}k_{3,6}k_{3,7}), \\
K[i][4] &= (k_{4,4}k_{4,5}k_{4,6}k_{4,7}), \\
K[i][5] &= (k_{5,5}k_{5,6}k_{5,7}k_{0,5}), \\
K[i][6] &= (k_{6,6}k_{6,7}k_{0,6}k_{1,6}), \\
K[i][7] &= (k_{7,7}k_{0,7}k_{1,7}k_{2,7}),
\end{aligned}$$

where K[i][j] is the binary representation of K_{ij}. The message $m = (m_0, m_1, \cdots, m_7)$ is then arranged as follows :

$$\begin{aligned}
M[0] &= (m_0m_1m_2m_3m_4), \\
M[1] &= (m_1m_2m_3m_4m_5), \\
M[2] &= (m_2m_3m_4m_5m_6), \\
M[3] &= (m_3m_4m_5m_6m_7), \\
M[4] &= (m_4m_5m_6m_7),
\end{aligned}$$

$$
\begin{aligned}
M[5] &= (m_5 m_6 m_7 m_0), \\
M[6] &= (m_6 m_7 m_0 m_1), \\
M[7] &= (m_7 m_0 m_1 m_2).
\end{aligned}
$$

The cipher-text of the i-th bit, $c[i]$, is computed in the following manner :

```
val=0;
for(j=0;j<8;j++)
  if(m[j]== 1)
     val=val^(K[i][j]^M[j]);
c[i]=hamming_weight(val) & 1;
```

where the function of hamming_weight(val) implies the Hamming weight of the binary representation of val. It should be noted that, the n bits of the cipher-text are computed on simultaneously as follows :

```
for(i=0;i<8;i++)
   val[i]=0;
for(j=0;j<8;j++)
  if(m[j]== 1)
     for(i=0;i<8;i++)
        val[i]=val[i]^(K[i][j]^M[j]);
for(i=0;i<8;i++)
   c[i]=hamming_weight(val[i]) & 1;
```

In this case, the average times of the word-based bit by bit exclusive OR operation equal 8 for i-th bit encryption and equal $64 = 8^2$ for the all of 8 bits encryption.

In general, when the n is smaller than 64, the average times of the word-based bit by bit exclusive OR operations equal n for the i-th bit encryption and equal n^2 for all of the n bits encryption. When the n is larger than 63, the number of bits of $K[i][j]$ and $M[i]$ is larger than 32. However rearranging the representations of $K[i][j]$ and $M[i]$, the average times of the word-based bit by bit exclusive OR operations equal $n \times_{n+1} C_2/32$ for the encryption. For example, if the word length equal 4 bits, the public key K_i is divided into 9 words so that

$$
\begin{aligned}
K[i][0] &= (k_{0,0} k_{0,1} k_{0,2} k_{0,3}), \\
\tilde{K}[i][0] &= (k_{0,4} k_{0,5} k_{0,6} k_{0,7}), \\
K[i][1] &= (k_{1,1} k_{1,2} k_{1,3} k_{1,4}), \\
K[i][2] &= (k_{2,2} k_{2,3} k_{2,4} k_{2,5}), \\
K[i][3] &= (k_{3,3} k_{3,4} k_{3,5} k_{3,6}), \\
K[i][4] &= (k_{4,4} k_{4,5} k_{4,6} k_{4,7}), \\
K[i][5] &= (k_{5,5} k_{5,6} k_{5,7} k_{1,5}), \\
K[i][6] &= (k_{6,6} k_{6,7} k_{1,6} k_{2,6}), \\
K[i][7] &= (k_{7,7} k_{1,7} k_{2,7} k_{3,7}),
\end{aligned}
$$

are satisfied.

When $n = 80$ and 100, the average times of the word-based bit by bit exclusive OR operations equal 102 and 158 respectively. The computational times described in 5. are the average times of the word-based bit by bit exclusive OR operations which are equal to 160 and 200 for $n = 80$ and 100 respectively. Therefore, the computational times can be reduced to $20\% \sim 30\%$ when the above rearrangement is used.