Efficient Variant of Rainbow without Triangular Matrix Representation

Takanori Yasuda, Tsuyoshi Takagi, Kouichi Sakurai

Efficient variant of Rainbow without triangular matrix representation

Takanori Yasuda[1], Tsuyoshi Takagi[2], and Kouichi Sakurai[1,3]

[1] Institute of Systems, Information Technologies and Nanotechnologies
[2] Institute of Mathematics for Industry, Kyushu University
[3] Department of Informatics, Kyushu University

Abstract. Multivariate Public Key Cryptosystems (MPKC) is one of candidates for post-quantum cryptography. Rainbow is an MPKC digital signature scheme, with relatively efficient encryption and decryption processes. However, the size of MPKC key is substantially larger than that of an RSA cryptosystem for the same security level. In this paper, we propose a variant of Rainbow that has a smaller secret key. The smaller secret key is to the result of a different description of the quadratic polynomials appearing in the secret key from that of the original Rainbow. In addition, our scheme improves the efficiency of the Rainbow's signature generation. In particular, the secret key is reduced in size by about 40% and the signature generation is sped up by about 30% at the security level of 100 bits.

Keywords: Post-quantum cryptography, Multivariate public key cryptosystems, Rainbow.

1 Introduction

Multivariate public key cryptosystems (MPKC) [1, 7] are candidates for post-quantum cryptography. Their security is based on the level of difficulty involved in finding solutions to a system of multivariate quadratic equations (MQ problem). Many MPKC schemes require secret and public keys that are larger than those of RSA and ECC. In recent years, a variety of MPKC schemes for encryption and for signatures, have been proposed. Unbalanced Oil and Vinegar (UOV) [5] is an MPKC signature scheme, whose signatures can be efficiently generated and verified. Rainbow [2] is a multilayer variant of UOV, with enhanced security. UOV and Rainbow both share the same problem of having large secret and public keys.

In this paper, we propose a variant of Rainbow that has a shorter secret key than the corresponding Rainbow key. In the case of the original Rainbow, the quadratic polynomials appearing in the secret key are expressed using triangular matrices. The non-zero parts of the triangular matrices coincide with coefficients of the quadratic polynomials. If we change the triangular matrices into general matrices, then the quadratic polynomials remain but, the correspondence of the matrix elements becomes more complicated. Conversely, if we utilize the

complicated correspondence, then simple matrix operation like rotation of row vectors yields several quadratic polynomials which seem to have been chosen randomly. Our scheme uses this method to describe the quadratic polynomials appearing in the secret key. In Rainbow, we need the same number of triangular matrices as that of quadratic polynomials, whereas in our scheme, we need only one matrix to describe the secret key.

Our scheme also improves the efficiency of signature generation. Here, we use several rotations of row vectors in a matrix so that the same matrix computation appears several times.

This paper analayzes the security of our scheme. In particular, we investigate the effect to our scheme for famous attacks against Rainbow, including direct attacks, HighRank attack and UOV attack. Among these attacks, we show that the complexities of the HighRank attack and UOV attack against our scheme are the same as those against the original Rainbow. Furthermore, we verify that the there is no difference in security between our scheme and Rainbow against direct attacks.

Finally, we evaluate the security parameter of our scheme for several security levels on the basis of our security analysis and the results in [6]. We also compare the secret key length and efficiency of signature generation of our scheme with those of the corresponding Rainbow. In particular, the size of the secret key of our scheme is reduced by about 40% and signature generation is about 30% faster at the security level of 100 bits.

2 Original Rainbow

Ding and Schmidt proposed a signature scheme called Rainbow, which is a multilayer variant of Unbalanced Oil and Vinegar [2].

First, we define parameters that determine the layer structure of Rainbow. Let t be the number of layers in Rainbow. Let $v_1, \ldots, v_{t+1}$ be a sequence of $t+1$ positive integers such that $0 < v_1 < v_2 < \cdots < v_t < v_{t+1}$. For $i = 1, \ldots, t$, the set of indices of the i-th layer in Rainbow is defined by all integers from v_i to v_{i+1}, namely $O_i = \{v_i + 1, v_i + 2, \ldots, v_{i+1} - 1, v_{i+1}\}$. The number of indices for the i-th layer, O_i is then $v_{i+1} - v_i$, and this is denoted by $o_i = v_{i+1} - v_i$. Note that the smallest integer in O_1 is $v_1 + 1$. Upon defining $V_1 = \{1, 2, \ldots, v_1\}$, and for $i = 2, 3, \ldots, t + 1$, we have

$$V_i = V_1 \cup O_1 \cup O_2 \cup \cdots \cup O_{i-1} = \{1, 2 \ldots, v_i\}.$$

The number of elements in V_i is exactly v_i for $i = 1, 2, \ldots, t + 1$. The sets O_i and V_i are used for the respective indices of the Oil and Vinegar variables in Rainbow. We define $n = v_{t+1}$ as the maximum number of variables used in Rainbow.

Next, let K be a finite field of order q. Rainbow consists of t layers of n variables polynomials. For $h = 1, 2, \ldots, t$, the h-th layer of Rainbow contains the

following system of o_h multivariate polynomials: For $k \in O_h$,

$$g_k(x_1, \ldots, x_n) = \sum_{i \in O_h, j \in V_h} \alpha_{i,j}^{(k)} x_i x_j + \sum_{i,j \in V_h,\, i \le j} \beta_{i,j}^{(k)} x_i x_j + \sum_{i \in V_{h+1}} \gamma_i^{(k)} x_i + \eta^{(k)}, \quad (1)$$

where $\alpha_{i,j}^{(k)}, \beta_{i,j}^{(k)}, \gamma_i^{(k)}, \eta^{(k)} \in K$. We call the variables x_i ($i \in O_h$) and x_j ($i \in V_j$) Oil and Vinegar variables, respectively. The central map of Rainbow is constructed according to $G = (g_{v_1+1}, \ldots, g_n) : K^n \to K^{n-v_1}$.

Note that a system of o_h equations, $g_k(b_1, \ldots, b_{v_h}, x_{v_h+1}, \ldots, x_{v_{h+1}}) = a_k$ ($k \in O_h$) becomes o_h linear equations in o_h variables for any $(a_{v_h+1}, \ldots, a_{v_{h+1}}) \in K^{o_h}$ and $(b_1, \ldots, b_{v_h}) \in K^{v_h}$. Therefore, once we know the values of the Oil variables in the h-th layer, we can then compute the values of the Vinegar variables in the $(h+1)$-th layer. This is the trapdoor mechanism of Rainbow.

2.1 Scheme

Now let us describe the key generation, signature generation, and verification processes of Rainbow.

Key Generation. The secret key consists of a central map G and two affine transformations $A_1 : K^m \to K^m$ ($m = n - v_1$), $A_2 : K^n \to K^n$. The public key consists of the field K and the composed map $F = A_1 \circ G \circ A_2 : K^n \to K^m$, which is a system of m quadratic polynomials of n variables over K. We denote the public key by $F = (f_{v_1+1}, \ldots, f_n)^{\mathrm{T}}$, where T denotes the transpose operation. In addition, we use f_k to denote the k-th public polynomial of F for $k = v_1 + 1, \ldots, n$.

Signature Generation. Let $\mathbf{M} \in K^m$ be a message. We compute $\mathbf{A} = A_1^{-1}(\mathbf{M})$, $\mathbf{B} = G^{-1}(\mathbf{A})$ and $\mathbf{C} = A_2^{-1}(\mathbf{B})$ in that order. The signature of the message is $\mathbf{C} \in K^n$. Note that the inverse of G can be efficiently computed. In fact, for any vector $w = (w_1, \ldots, w_m)^T \in K^m$, an element $G^{-1}(w)$ in the inverse image of w can be obtained as follows:

Step 1 Randomly choose $s'_1, \ldots, s'_{v_1} \in K$.
Step 2 For $i = 1, \ldots, t$, do the following operations:
A system $g^{(v_i+1)}, \ldots, g^{(v_i+o_i)}$ can be regarded as a multivariate quadratic system with variables $x_1, \ldots, x_{v_i+o_i}$. Upon substituting $(x_1, \ldots, x_{v_i}) = (s'_1, \ldots, s'_{v_i})$, set up a system of linear equations of o_i variables. Solve the system and obtain a solution $(x_{v_i+1}, \ldots, x_{v_i+o_i}) = (s'_{v+1}, \ldots, s'_n)$. (If the system is not regular, go back to Step 1.)
Result $G^{-1}(w) = (s'_1, \ldots, s'_n)$.

Verification. If $F(\mathbf{C}) = \mathbf{M}$, the signature is accepted; it is rejected otherwise.

This scheme is denoted as Rainbow(K; $v_1, o_1, \ldots, o_t$), and we call $v_1, o_1, \ldots, o_t$ the parameters of Rainbow.

3 A variant of Rainbow

In this section, we present our variant of Rainbow, called matrix-based Rainbow. Our scheme uses a special secret key to improve Rainbow's signature generation algorithm.

3.1 Basic Underlying Idea

We focus on the terms

$$\sum_{i,j \in S_h,\, i \leq j} \beta_{i,j}^{(k)} x_i x_j \tag{2}$$

appearing in the components g_k of the quadratic polynomial map G that composes the secret key of Rainbow. Using a square matrix of size v_h,

$$B = \begin{pmatrix} \beta_{1,1}^{(k)} & \beta_{1,2}^{(k)} & \cdots & \beta_{1,v_h}^{(k)} \\ 0 & \beta_{2,2}^{(k)} & * & \vdots \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \beta_{v_h,v_h}^{(k)} \end{pmatrix},$$

the terms can be expressed as

$$\mathbf{x}.B.\mathbf{x}^T \quad (\mathbf{x} = (x_1, \ldots, x_{v_h})). \tag{3}$$

Let us see the change in (3) that is had by replacing the triangular matrix B with a general matrix. For a general matrix,

$$D = \begin{pmatrix} \delta_{1,1} & \cdots & \cdots & \delta_{1,v_h} \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & \vdots \\ \delta_{v_h,1} & \cdots & \cdots & \delta_{v_h,v_h} \end{pmatrix},$$

$\mathbf{x}.D.\mathbf{x}^T$ is expressed by

$$\sum_{i,j \in S_h,\, i<j} (\delta_{i,j} + \delta_{j,i})\, x_i x_j + \sum_{i \in S_h} \delta_{i,i}\, x_i^2. \tag{4}$$

Comparing this with (2), it is clear that the terms are more complicated. In addition, we will rotate down the row vectors in the matrix D by l rows; that is, for

$$D_l = \begin{pmatrix} \delta_{v_h-l+1,1} & \cdots & \cdots & \delta_{v_h-l+1,v_h} \\ \delta_{v_h-l+2,1} & \cdots & \cdots & \delta_{v_h-l+2,v_h} \\ \vdots & & & \vdots \\ \delta_{v_h-l,1} & \cdots & \cdots & \delta_{v_h-l,v_h} \end{pmatrix},$$

$\mathbf{x}.D_l.\mathbf{x}^T$ can be expressed as

$$\sum_{i,j \in S_h,\, i<j} (\delta_{i-l,j} + \delta_{j-l,i})\, x_i x_j + \sum_{i \in S_h} \delta_{i-l,i}\, x_i^2. \tag{5}$$

Here, the indices are regarded as numbers modulo v_h.

Since number of the matrices in which row vectors are rotated is v_h, at most v_h quadratic polynomials of the form $\mathbf{x}.D_l.\mathbf{x}^T$ are obtained. As we can see from the coefficients of (5), it seems difficult to relate the coefficients in the form not appeared by $\delta_{i,j}$. In other words, from a general matrix D, we can construct up to v_h quadratic polynomials that look independent of each other.

The method is used to construct the terms (2) appearing in the secret key of our scheme. More concretely, for the h-th layer, we prepare a matrix D, and construct the terms (2) appearing in $g_{v_h+1}, \ldots, g_{v_{h+1}}$ by rotating the rows of D.

In the original Rainbow, $o_h(= v_{h+1} - v_h)$ triangular matrices are needed to describe the secret key, whereas in our scheme, only one matrix D is needed. Comparing these parts, we find that the number of elements in the base field of our scheme is reduced by $2/o_h$. Therefore, its secret key is shorter. We should remark that since the number of rows in D is at most v_h, for any h-th layer, the condition $v_h \geq o_h$ has to be satisfied for the design to be secure.

3.2 Construction of the Secret Key

In our scheme, we use the invertible map G used in the original Rainbow as a trapdoor. However, the choice of the coefficients of G in our scheme is different from that in the original Rainbow.

$t, v_1, \ldots, v_{t+1}, m, n$ are natural numbers satisfying the same condition as in § 2.1. In addition, for any $h = 1, \ldots, t$, it is assumed that $v_h \geq o_h$. For $i = 1, \ldots, t$, we write $S_i = \{1, \ldots, v_i\}$, $O_i = \{v_i + 1, \ldots, v_{i+1}\}$, and $o_i = v_{i+1} - v_i$.

First, for each $h = 1, \ldots, t$, we choose a square matrix $D^{(h)} = (\delta_{i,j}^{(h)})$, with a size v_h. Then, we create a map $G = (g_{v_1+1}, \ldots, g_n) : K^n \to K^m$ consisting of quadratic polynomials in the following form: For $h = 1, \ldots, t,\ l = 1, \ldots, o_h$, we write $k = v_h + l$, and

$$g_k(x_1, \ldots, x_n) = \sum_{i \in O_h, j \in S_h} \alpha_{i,j}^{(k)} x_i x_j + \sum_{i \in S_h} \delta_{i-l+1,i}^{(h)} x_i^2 + \sum_{i,j \in S_h,\, i<j} (\delta_{i-l+1,j}^{(h)} + \delta_{j-l+1,i}^{(h)})\, x_i x_j + \sum_{i \in S_{h+1}} \gamma_i^{(k)} x_i + \eta^{(k)}. \tag{6}$$

Here, $\alpha_{i,j}^{(k)}, \gamma_i^{(k)}, \eta^{(k)} \in K$.

3.3 Efficient Inverse Computation

Since the map G defined above can be expressed in the form used in the definition of Rainbow, we use the algorithm for computing the inverse map of G that is used in the signature generation of the original Rainbow. Moreover, we can improve the algorithm.

In the signature generation of the original Rainbow, the linear system appearing in the inverse map computation of G is described as $L.X = V$ where,

$$L = (\sum_{k=1}^{v_h} \alpha_{k,v_h+j}^{(v_h+i)} b_k + \gamma_{v_h+j}^{(v_h+i)})_{1\leq i,j\leq o_h}, \tag{7}$$

$$V = \left(a_{v_h+i} - \sum_{1\leq k\leq l\leq v_h} \beta_{k,l}^{(v_h+i)} b_k b_l - \sum_{1\leq k\leq v_h} \gamma_k^{(v_h+i)} b_k - \eta^{(v_h+i)} \right)_{1\leq i\leq o_h}, \tag{8}$$

$$X = (x_{v_h+1}, \ldots, x_{v_{h+1}})^T.$$

In our scheme, that is

$$\beta_{i,j}^{(v_h+l)} = \begin{cases} \delta_{i-l+1,j}^{(h)} + \delta_{j-l+1,i}^{(h)} & i \neq j \\ \delta_{i-l+1,i}^{(h)} & i = j. \end{cases}$$

Accordingly, $\sum_{1\leq k\leq l\leq v_h} \beta_{k,l}^{(v_h+i)} b_k b_l$ appearing in V is expressed as $\mathbf{b}.D_{i-1}^{(h)}.\mathbf{b}^T$ ($\mathbf{b} = (b_1, \ldots, b_{v_h})$). Here, $D_{i-1}^{(h)}$ is a matrix whose row vectors are rotated down from those of $D^{(h)}$ by $i-1$ rows. For any $i = 1, \ldots, o_h$, $D_{i-1}^{(h)}.\mathbf{b}^T$ coincides with a column vector rotated down from the components of $D^{(h)}.\mathbf{b}^T$ by $i-1$ elements. Therefore, if we compute $D^{(h)}.\mathbf{b}^T$ once, $D_{i-1}^{(h)}.\mathbf{b}^T$ does not have to be computed. Computing the inner form of this and $\mathbf{b}$, $\sum_{1\leq k\leq l\leq v_h} \beta_{k,l}^{(v_h+i)} b_k b_l$ can be computed. In our scheme, this method is used to compute the inverse of G is computed.

3.4 Our Scheme

Our scheme uses the invertible map G for the key generation, signature generation and verification as follows:

• **Key generation**

Secret key The secret key consists of the quadratic map G, and two randomly chosen affine transformations $L : K^m \to K^m$ and $R : K^n \to K^n$.

Public key The public key consists of the composite map $F = L \circ G \circ R : K^n \to K^m$.

• **Signature generation** Let $\mathbf{M} \in K^m$ be a message. To generate a signature $\mathbf{S}$ from $\mathbf{M}$, first compute $\mathbf{M}' = L^{-1}(\mathbf{M})$. Next compute an element $\mathbf{S}' = G^{-1}(\mathbf{M}')$ in the inverse image of $\mathbf{M}'$, and finally compute $\mathbf{S} = R^{-1}(\mathbf{S}')$. $G^{-1}(\mathbf{M}')$ is computed using the improved algorithm described above. $L^{-1}(\mathbf{M})$ and $R^{-1}(\mathbf{S}')$ can be easily computed since L and R are affine transformations, .

• **Verification** If $F(\mathbf{S}) = \mathbf{M}$, the signature is accepted. It is rejected otherwise.

We denote this scheme by NT-Rainbow$(K; v_1, d_1 * o_1', \ldots, d_t * o_t')$ and call $v_1, d_1, o_1', \ldots, d_t, o_t'$ the parameter.

4 Security Analysis for Our Scheme

Now let us analyze the security of our scheme against several attacks.

4.1 Security against Direct Attacks

We experimentally compared the time taken by direct attacks against our scheme NT-Rainbow($GF(256); v_1, o_1, o_2$) over against the time taken by the same attack against Rainbow($GF(256); v_1, o_1, o_2$). The experiment used the gröbner basis implemented in Magma. The table below lists the results: It shows that there is no significant difference between the times of those schemes.

Table 1. Comparison of Time Taken by Direct Attacks over $GF(256)$

(v_1, o_1, o_2)	(4,3,4)	(5,3,4)	(3,4,4)
Our scheme	5.32 s	11.70 s	13.80 s
Rainbow	5.34 s	11.70 s	13.84 s
Random system	5.36 s	11.72 s	13.88 s

4.2 Security against HighRank Attack

We can write $g^{(2)}_{v_1+1}, \ldots, g^{(2)}_n$ for the quadratic parts of the components of the central map $G = (g_{v_1+1}, \ldots, g_n)$. Each $g^{(2)}_i$ is expressed by $g^{(2)}_i(\mathbf{x}) = \mathbf{x}.T_i.\mathbf{x}^T$, $(\mathbf{x} = (x_1, \ldots, x_n))$ using a triangular matrix T_i of size n. The symmetric matrix S_i $(i = v_1 + 1, \ldots, n)$ is defined by $S_i = T_i + T_i^T$, and we can write $\mathcal{A} = \mathrm{Span}_K\{S_{v_1+1}, \ldots, S_n\}$.

The HighRank attack finds a matrix in $\mathcal{A}$ with the maximal rank (not full rank), and it spends most of its times in this process. The computation has the following steps.

Step 1 Choose $M \in \mathcal{A}$ randomly.
Step 2 Determine whether M is regular. If M is regular, then return to Step 1
Output M.

The complexity of the computation for determining whether M is regular or not (which is equal to complexity of HighRank attack) is $q^{o_t}n^3/6$ field multiplication([3],[6]). If the $\alpha^{(k)}_{i,j}$'s are chosen randomly, the probability that M is not regular is equal to $1/q^{o_t}$. In our scheme, $\alpha^{(k)}_{i,j}$'s are chosen randomly as in the original Rainbow. Therefore, the complexity of the HighRank attack against our scheme is equivalent to that against the original Rainbow.

Security against UOV Attack The space spanned by the variables $x_{v_t+1}, \ldots, x_n$ is a simultaneously isotropic space with respect to $g^{(2)}_{v_1+1}, \ldots, g^{(2)}_n$. Here, a subspace W of a vector space V with a quadratic form g is said to be isotropic if $v_1, v_2 \in V \Rightarrow g(v_1, v_2) = 0$.

The UOV attack finds the simultaneously isotropic space by using the following steps.

Step 1 Randomly choose $M_1, M_2 \in \mathcal{A}$ such that M_2 is regular.
Step 2 Compute a proper invariant subspace W of $M_{1,2} = M_1 M_2^{-1}$. If there is no invariant subspace, return to Step 1.
Output W.

The complexity of the UOV attack is $q^{n-2o_t-1}o_t^3$ field multiplication([4]). If the $\alpha_{i,j}^{(k)}$'s are chosen randomly, the probability that $M_{1,2}$ has an invariant subspace is equal to $1/q^{n-2o_t}$. In our scheme, the $\alpha_{i,j}^{(k)}$'s are chosen randomly similarly as in the original Rainbow. Therefore, the complexity of UOV attack against our scheme is equivalent to that against the original Rainbow.

5 Efficiency of Signature Generation

In this section, we compare the efficiencies of signature generation of our scheme and the original Rainbow. The respective schemes are fixed by setting NT-Rainbow(K; $v_1, o_1, \ldots, o_t$) and Rainbow($K; v_1, o_1, \ldots, o_t$). The previous section shows these schemes have the same security level against direct attacks, the HighRank attack and the UOV attack.

5.1 Efficiency of Signature Generation

Now let us estimate the number of multiplications and additions of K in the signature generation in our scheme and the original Rainbow. We fix the schemes by setting Rainbow($K; v_1, o_1, \ldots, o_t$) and NT-Rainbow(K; $v_1, o_1, \ldots, o_t$) and choose Gaussian elimination as the solver of the linear systems appearing in the signature generation. For both cases of the original Rainbow and our scheme, we consider to generate a signature corresponding to a message $\mathbf{M} \in K^m$. Then, we have to compute $\mathbf{A} = L^{-1}(\mathbf{M})$, $\mathbf{B} = G^{-1}(\mathbf{A})$ and $\mathbf{C} = R^{-1}(\mathbf{B})$ in this order and obtain a signature $\mathbf{C}$. The computation of $\mathbf{A} = L^{-1}(\mathbf{M})$ and $\mathbf{C} = R^{-1}(\mathbf{B})$ are common for both schemes.

The respective costs for computing $\mathbf{B} = G^{-1}(\mathbf{A})$ are described as follows:

Original Rainbow

$$\text{Multiplication} \quad \sum_{h=1}^{t} \left(\frac{o_h v_h^2}{2} + \frac{o_h^3}{3} + (v_h+1)o_h^2 + \frac{3o_h v_h}{2} + \frac{v_h(v_h+1)}{2} - \frac{o_h}{3} \right),$$

$$\text{Addition} \quad \sum_{h=1}^{t} \left(\frac{o_h v_h^2}{2} + \frac{o_h^3}{3} + (v_h+1)o_h^2 + \frac{3o_h v_h}{2} - \frac{o_h}{3} \right).$$

Our scheme

$$\text{Multiplication} \quad \sum_{h=1}^{t} \left(v_h o_h^2 + \frac{o_h^3}{3} + v_h^2 + 2o_h v_h + o_h^2 - \frac{o_h}{3} \right),$$

$$\text{Addition} \quad \sum_{h=1}^{t} \left(v_h o_h^2 + \frac{o_h^3}{3} + v_h^2 + 2o_h v_h + o_h^2 - v_h - \frac{o_h}{3} \right).$$

Comparison of Efficiencies The term $v_h o_h^2$ appears in the cost computation of the original Rainbow, but not in our scheme. Moreover, the cubic terms with respect to v_h, o_h in the rest of the equation except for $v_h o_h^2$ are almost the same. This means the signature generation of our scheme is more efficient than that of the original Rainbow.

6 Examples

Our scheme NT-Rainbow$(K; v_1, o_1, \ldots, o_t)$ is valid if $v_h \geq o_h$ for all $h = 1, \ldots, t$. The security analysis of § 4 indicates that this scheme with this parameter must have same security level as that of Rainbow$(K; v_1, o_1, \ldots, o_t)$ against direct attacks, the HighRank attack and UOV attack. Petzoldt et al. [6] describe the corresponding parameters of the original Rainbow for several security levels. Since these parameters satisfy $v_h \geq o_h$ for all $h = 1, \ldots, t$, our schemes with the corresponding parameters exist. Table 2 lists the parameters of our schemes for security levels of 80, 90, 100-bits in correspondence with the parameters of Petzoldt et al. Our scheme NT-Rainbow$(K; v_1, o_1, o_2)$ is one with two-layers over $K = GF(256)$. Next, we compare the secret key lengths and the efficiencies of

Table 2. Parameters of Our Scheme over $GF(256)$ and its Security Level

Parameter (v_1, o_1, o_2)	$(18, 14, 14)$	$(24, 17, 18)$	$(31, 21, 22)$
Security Level	80 bits	90 bits	100 bits

the signature generation of our scheme and the original Rainbow for these parameters. Table 3 compares the secret key lengths, and Table 4 compares the efficiencies of the signature generation. Table 4 shows the number of the multiplications and additions of $GF(256)$, and the time taken by a C-Language implementation. We used gcc and an Intel Core i5 2.67GHz CPU with 4GB RAM.

Table 3. Secret Key Lengths of Schemes over $GF(256)$

Parameter (v_1, o_1, o_2)	$(18, 14, 14)$	$(24, 17, 18)$	$(31, 21, 22)$
Security Level	80 bits	90 bits	100 bits
Secret Key Length of Our Scheme(Byte)	15284	29071	52709
Secret Key Length of Rainbow(Byte)	23680	47412	89776
Ratio	64.6%	61.3%	58.7%

Table 4. Efficiencies of Signature Generation of Schemes over $GF(256)$

Parameter (v_1, o_1, o_2)	$(18, 14, 14)$	$(24, 17, 18)$	$(31, 21, 22)$
Signature Generation	Mult:17660, Add:17536	Mult:33658, Add:33499	Mult:60966, Add:60766
Experiment(μs)	144	270	475
Signature Generation	Mult:26097, Add:25324	Mult:52014, Add:50759	Mult:98112, Add:96121
Experiment(μs)	189	370	695
Ratio	Mult:67.7%, Add:69.2%	Mult:64.7%, Add:63.5%	Mult:62.1%, Add:63.2%
	76.2%	73.0%	68.3%

7 Conclusion

We presented a variant of Rainbow, that has a smaller secret key and faster signature generation process compared with the original. We analyzed the security of our scheme against known attacks such as direct attacks, HighRank attack, and UOV attack. In addition, we presented an explicit parameter of our scheme for several security levels. Our test proves that our scheme is 30% faster than Rainbow at generating the signatures and has a 40% smaller key at a security level of 100 bits.

Acknowledgments

This work was supported by "Strategic Information and Communications R&D Promotion Programme (SCOPE), no. 0159-0172", Ministry of Internal Affairs and Communications, Japan. The first author is supported by Grant-in-Aid for Young Scientists (B), Grant number 24740078.

References

1. Ding, J., Gower, J. E. and Schmidt, D. S., "Multivariate Public Key Cryptosystems", Advances in Information Security 25, Springer, 2006.
2. Ding, J. and Schmidt, D., "Rainbow, a New Multivariable Polynomial Signature Scheme", ACNS'05, Springer LNCS vol. 3531, pp. 164–175, 2005.
3. Ding, J. Yang, B.-Y., Chen, C.-H. O., Chen, M.-S. and Cheng, C. M., "New Differential-Algebraic Attacks and Reparametrization of Rainbow", ACNS'08, Springer LNCS vol. 5037, pp. 242–257, 2008. CANS'05, Springer LNCS vol. 3810, pp. 211–222, 2005.
4. Kipnis, A., Patarin, J. and Goubin, L., "Unbalanced Oil and Vinegar Schemes", EUROCRYPT'99, Springer LNCS vol. 1592, pp. 206–222, 1999.
5. Patarin, J., "The Oil and Vinegar Signature Scheme", Dagstuhl Workshop on Cryptography, 1997.
6. Petzoldt, A., Bulygin, S. and Buchmann, J., "Selecting Parameters for the Rainbow Signature Scheme", PQCrypto'10, Springer LNCS vol. 6061, pp. 218–240, 2010.
7. Wolf, C., "Introduction to Multivariate Quadratic Public Key Systems and their Applications", YACC'06, pp. 44–55, 2006.