A Variant of Rainbow with Shorter Secret Key and Faster Signature Generation

Takanori Yasuda
Institute of Systems, Information Technologies and Nanotechnologies
yasuda@isit.or.jp

Jintai Ding
Chongqing University
University of Cincinnati
jintai.ding@uc.edu

Tsuyoshi Takagi
Institute of Mathematics for Industry, Kyushu University
takagi@imi.kyushu-u.ac.jp

Kouichi Sakurai
Institute of Systems, Information Technologies and Nanotechnologies
Department of Informatics, Kyushu University
sakurai@csce.kyushu-u.ac.jp

ABSTRACT

Multivariate public key cryptosystems are being focused on as candidates for post-quantum cryptography. Rainbow is one of the most efficient signature schemes in multivariate public key cryptosystems. The main drawback of Rainbow is that their key size is much larger than that of RSA and ECC. In this paper, we propose an efficient variant of Rainbow that has a shorter secret key (and thus generates signatures faster) than the corresponding original Rainbow. In our scheme, we divide each layer of Rainbow into smaller blocks by using diagonal matrix representations. The size of the smaller blocks can be flexibly selected, and this enables us to carefully choose secure parameters so that our proposed scheme is secure against known attacks such as rank attacks, direct attacks, and UOV attack. We estimate that the secret key size of our proposed scheme with 100-bit security is smaller by about 40% than that of the original Rainbow. In addition, an implementation of our scheme in the C language is seen to generate signature faster by 40%.

Categories and Subject Descriptors

E.3 [**DATA ENCRYPTION**]: Public key cryptosystems

General Terms

Theory

AsiaPKC'13, May 8, 2013, Hangzhou, China.

Keywords

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

1. INTRODUCTION

Multivariate Public Key Cryptosystems (MPKC) are candidates of post-quantum cryptography. Their security depends on the difficulty of solving a system of quadratic equations of multivariables. Rainbow [4] is a signature scheme in MPKC, whose signatures can be efficiently generated and verified. In general, an MPKC scheme needs a huge public key for security. In addition, Rainbow also have huge secret keys. In fact, Rainbow with 80-bit security level has secret and public keys that are more than 150 times as large as those of 1024-bit RSA [10]. Therefore an important research goal is to find a way to reduce the sizes of the secret and public keys of Rainbow.

In fact, many papers reducing the key sizes have been published. Some methods have been summarized by Wolf et al. [14]. Secret keys of MPKC can be reduced by using sparse polynomials for the central map. TTS [16] is a typical example of doing so. Yasuda et al. reduced the size of secret key of Rainbow by using a compact regular representation with non-commutative rings [17]. CyclicRainbow is a variant of UOV whose public key size can be reduced by repeated usage of a sequence of coefficients in polynomials in the public key [11]. By an application of technique of CyclicRainbow, a scheme in which many coefficients of public key are 0 or 1 was proposed [12].

In this paper, we propose an extension of Rainbow, called matrix-based Rainbow. The proposed scheme provides a new way to reduce secret the key size of Rainbow. The initial inspiration behind them comes from "Rainbow using non-commutative rings" in [17]. The main idea of our scheme is compressing systems of linear equations appearing in the signature generation. The original Rainbow generates signatures by solving large systems of linear equations with respect to oil variables. In our scheme, on the other hand,

requires one to solve a much smaller the systems of linear equations. Accordingly, it reduces the size of the secret key and speed up the signature generation in comparison with the corresponding original Rainbow. We analyse the security of our scheme against known attacks against the original Rainbow, including rank attacks, direct attacks, UOV attack, and so on. We then discuss that security is not weakened against these attacks. In the case of matrix-based Rainbow with 100-bit security, the size of the secret key is reduced by about 40% and the signature generation is sped up by about the same amount. We implemented the proposed matrix-based Rainbow in C on a Core i5, and the experiment shows that its signature generation is about 34% faster than that of the corresponding Rainbow of 100-bit security.

2. RAINBOW

In this section, we give a short overview of a signature scheme, Rainbow. Ding and Schmidt [4] proposed the original Rainbow signature scheme to which invertible multivariate quadratic systems used in UOV signature scheme is applied. Let v_1, $o_1, \ldots, o_t$ be $t+1$ positive integers. We write $v_i = v_1 + \sum_{j<i} o_j$ for $i = 2, \ldots, t$. The number of equations and variables in the multivariate quadratic system used in the Rainbow described here is $m = \sum_{i=1}^{t} o_i$ and $n = m + v_1$, respectively. Let $G = (g^{(v_1+1)}, \ldots, g^{(n)})$ be a map from K^n to K^m where each $g^{(h)}$ $(v_1 + 1 \le h \le n)$ is a quadratic polynomial of the form

$$g^{(h)}(\mathbf{x}) = \mathbf{x}^T A^{(h)} \mathbf{x} + B^{(h)} \mathbf{x} + C^{(h)}, \quad (\mathbf{x} = (x_1, \ldots, x_n)^T).$$

Here, $A^{(h)}$ is a square matrix over K with size n expressed by

$$A^{(v_i+j)} = \left(\begin{array}{c|c} A_0^{(v_i+j)} & 0 \\ \hline 0 & 0 \end{array}\right) \quad (i = 1, \ldots, t,\ j = 1, \ldots, o_i),$$

where $A_0^{(v_i+j)}$ $(j = 1, \ldots, o_i)$ are square matrices with size v_{i+1} of the form

$$A_0^{(v_i+j)} = \left(\begin{array}{c|c} A_{00}^{(v_i+j)} & A_{01}^{(v_i+j)} \\ \hline 0 & 0 \end{array}\right)$$

where $A_{00}^{(v_i+j)}$ is a randomly chosen upper triangular square matrix with size v_i and $A_{01}^{(v_i+j)}$ is a randomly chosen $v_i \times o_i$ - matrix. $B^{(h)}$ is a vector in K^n expressed in the form,

$$B^{(v_i+j)} = (B_0^{(v_i+j)}, \overbrace{0, \ldots, 0}^{n-v_{i+1}}) \quad (i = 1, \ldots, t,\ j = 1, 2, \ldots, o_i),$$

where $B_0^{(v_i+j)}$ is a randomly chosen vector in $K^{v_{i+1}}$. $C^{(h)}$ is a randomly chosen element in K. The inverse of map 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 is 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 operation:
A system $g^{(v_i+1)}, \ldots, g^{(v_i+o_i)}$ can be regarded as an invertible multivariate quadratic system with variables $x_1, \ldots, x_{v_i+o_i}$. 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 $G^{-1}(w) = (s'_1, \ldots, s'_n)$.

Using the invertible map G, the key generation, signature generation, and verification of Rainbow are described as follows:

• **Key generation**

Secret key The secret key consists of the map $G : K^n \to K^m$, 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 then finally compute $\mathbf{S} = R^{-1}(\mathbf{S}')$. The computation algorithm for $G^{-1}(\mathbf{M}')$ has already been explained. Since L and R are affine transformations, $L^{-1}(\mathbf{M})$ and $R^{-1}(\mathbf{S}')$ can be easily computed.

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

This scheme is denoted by Rainbow$(K; v_1, o_1, \ldots, o_t)$ and we call $v_1, o_1, \ldots, o_t$ a parameter of Rainbow, and t the number of layer of Rainbow.

3. MATRIX-BASED RAINBOW

In this section, we present our variant of Rainbow, called matrix-based Rainbow. Our scheme uses a special secret key of Rainbow to improve Rainbow's signature generation algorithm. We will start by explaining the basic idea underlying our scheme in the context of Rainbow with 1 layer.

3.1 Basic Underlying Idea

The key idea underlying our scheme is a modification of linear equations appearing in Step 2 of the Rainbow signature generation process. We assume that the Rainbow has 1 layer and is described by Rainbow$(K; v, o)$. In Step 2 of the Rainbow signature generation process, we need to solve a system of linear equations described as

$$L.X = V \tag{1}$$

where L is a $o \times o$-matrix, V is a column vector of size o and X is a column vector of variables of size o. First, we change vector V on the right hand side of the equation. Assume that o is factored as $o = do'$. V can be divided into d partitions consisting of o' elements in a natural way. Therefore, the o dimensional vector V corresponds to a matrix V' of size $o' \times d$ by the following correspondence:

$$V = \begin{pmatrix} \eta_1 \\ \eta_2 \\ \vdots \\ \eta_o \end{pmatrix} \mapsto V' = \begin{pmatrix} \eta_{11} & \eta_{12} & \cdots & \eta_{1d} \\ \eta_{21} & \eta_{22} & \cdots & \eta_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ \eta_{o'1} & \eta_{o'2} & \cdots & \eta_{o'd} \end{pmatrix}.$$

Similarly, we can change the vector $X = (x_{v+1}, \ldots, x_{v+o})^T$ of o variables in (1), i.e. a matrix X' with variables of size $o' \times d$ is assigned to the o dimensional vector X by

$$X = \begin{pmatrix} x_{v+1} \\ x_{v+2} \\ \vdots \\ x_{v+o} \end{pmatrix} \mapsto X' = \begin{pmatrix} x_{v+1} & x_{v+o'+1} & \cdots & x_{v+o'(d-1)+1} \\ x_{v+2} & x_{v+o'+2} & \cdots & x_{v+o'(d-1)+2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{v+o'} & x_{v+2o'} & \cdots & x_{v+o'd} \end{pmatrix}.$$

In addition, we want to change matrix L of size $o \times o$ in (1) into a matrix $L' = (\gamma_{ij})$ of size $o' \times o'$ such that equation (1) becomes equivalent to the following equation:

$$L'.X' = V'. \tag{2}$$

Generally, this change is impossible. However, if L is expressed as

$$L = \left(\begin{array}{c|c|c|c} A & 0 & \cdots & 0 \\ \hline 0 & A & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & A \end{array}\right) \tag{3}$$

for some matrix A of size $o' \times o'$, where each block size is $o' \times o'$, equation (1) becomes equivalent to equation (2) for $L' = A$.

We call equations in the form of (2) equations of matrix type of size $o' \times d$. Our scheme uses equations of the matrix type in stead of equations (1). System of equations of the matrix type can be solved simultaneously with respect to the columns of variables in X'. If Gaussian elimination is used to solve a system of linear equations, the occurrences of field multiplications and additions are estimated to be $O({o'}^3)$. In general, a system of equations of the matrix type of size $o' \times d$ can be solved more efficiently than a system of equations of the usual type with o variables because the complexity of solving a system of the usual type is $O(o^3)$, and $o' < o$. Therefore, a scheme based on the above idea is more efficient at signature generation than the original Rainbow.

3.2 Special Secret Key of Rainbow

We propose a variant of Rainbow based on the idea outlined in the last subsection. Our scheme requires special secret key of Rainbow, and let us explain how our scheme's secret key is generated.

Let $v_1, o_1, \ldots, o_t$ be $t+1$ positive integers, as in § 2. We write $v_i = v_1 + \sum_{j<i} o_j$ for $i = 2, \ldots, t$. The scheme which we will describe from now on is a variant of Rainbow$(K; v_1, o_1, \ldots, o_t)$, which was described in § 2. The number of variables and equations in the multivariate quadratic system used in the scheme is $m = \sum_{i=1}^{t} o_i$ and $n = m + v_1$, respectively.

Assume that for all $i = 1, \ldots, t$, o_i can be factored as $o_i = d_i o_i'$ for some positive number o_i', d_i. We first randomly generate the following matrices and vectors over K: For all $i = 1, \ldots, t$,

1. $\mathbf{a}_j^{(i)}$: $v_i \times o_i'$-matrix over K $\quad (j = 1, \ldots, o_i')$,
2. $\mathbf{b}_j^{(i)} \in K^{o_i'}$ $\quad (j = 1, \ldots, o_i')$.

We define a quadratic map $G : K^n \to K^m$ as a special form of the quadratic map G defined in § 2 for the original Rainbow. In our scheme, the description of $A_{01}^{(h)}$ and $B_0^{(h)}$ of G for the original Rainbow are specified. More specifically, a quadratic map $G : K^n \to K^m$ for our scheme is described as follows: $G = (g^{(v_1+1)}, \ldots, g^{(n)})$ is composed of quadratic polynomials $g^{(h)}$ of the form

$$g^{(h)}(\mathbf{x}) = \mathbf{x}^T A^{(h)} \mathbf{x} + B^{(h)} \mathbf{x} + C^{(h)}, \quad (\mathbf{x} = (x_1, \ldots, x_n)^T). \tag{4}$$

Here, $A^{(h)}$ is a square matrix over K with size n expressed by

$$A^{(v_i+j)} = \left(\begin{array}{c|c} A_0^{(v_i+j)} & 0 \\ \hline 0 & 0 \end{array}\right) \qquad (i = 1, \ldots, t,\ j = 1, \ldots, o_i),$$

where $A_0^{(v_i+j)}$ $(j = 1, \ldots, o_i)$ are square matrices with size v_{i+1} of the form

$$A_0^{(v_i+j)} = \left(\begin{array}{c|c} A_{00}^{(v_i+j)} & A_{01}^{(v_i+j)} \\ \hline 0 & 0 \end{array}\right)$$

where $A_{00}^{(v_i+j)}$ is a randomly chosen upper triangular square matrix with size v_i and $A_{01}^{(v_i+j)}$ is a $v_i \times o_i$-matrix defined by

$$A_{01}^{(v_i+ho_i'+r)} = (\overbrace{\mathbf{0}, \ldots, \mathbf{0}}^{ho_i'}, \mathbf{a}_r^{(i)}, \overbrace{\mathbf{0}, \ldots, \mathbf{0}}^{(d_i-h-1)o_i'}) \quad (0 \le h < d_i, 0 < r \le o_i').$$

($\mathbf{0}$ represents a column vector.) $B^{(h)}$ is a vector in K^n expressed in the form,

$$B^{(v_i+j)} = (B_0^{(v_i+j)}, \overbrace{0, \ldots, 0}^{n-v_{i+1}}) \quad (i = 1, \ldots, t, j = 1, 2, \ldots, o_i).$$

Here, $B_0^{(v_i+j)}$ is a vector in $K^{v_{i+1}}$ given by

$$B_0^{(v_i+j)} = (B_{00}^{(v_i+j)}, B_{01}^{(v_i+j)})$$

where $B_{00}^{(v_i+j)}$ is any vector in K^{v_i} and $B_{01}^{(v_i+j)} \in K^{o_i}$ is defined by

$$B_{01}^{(v_i+ho_i'+r)} = (\overbrace{0, \ldots, 0}^{ho_i'}, \mathbf{b}_r^{(i)}, \overbrace{0, \ldots, 0}^{(d_i-h-1)o_i'}) \quad (0 \le h < d_i, 0 < r \le o_i').$$

$C^{(h)}$ is a randomly chosen element in K.

3.3 Inverse Computation

Since our G is a special form of the G given in § 2, the inverse of G can also be computed using the algorithm described in § 2. Our special construction of G enables us to improves the efficiency of the computation. Let us now look at how the algorithm is modified.

In Step 2, in the algorithm of the inverse of G in § 2, several systems of linear equations appear. These systems are described as

$$L.X = V. \tag{5}$$

In the case of our G, L can be described using a blockwise diagonal matrix

$$L = \left(\begin{array}{c|c|c|c} A & 0 & \cdots & 0 \\ \hline 0 & A & \cdots & 0 \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline 0 & 0 & \cdots & A \end{array}\right), \quad \left(A = \begin{pmatrix} \mathbf{s}'.\mathbf{a}_1^{(i)} \\ \vdots \\ \mathbf{s}'.\mathbf{a}_{o_i'}^{(i)} \end{pmatrix} + \begin{pmatrix} \mathbf{b}_1^{(i)} \\ \vdots \\ \mathbf{b}_{o_i'}^{(i)} \end{pmatrix}\right)$$

for $\mathbf{s}' = (s_1', \ldots, s_{v_i}')$. From the observation in § 3.1, system (5) can be transformed into a system (2) of linear equations of the matrix type. This system of the matrix type can be computed more efficiently than system (5).

3.4 Our Scheme

Using the invertible map G, the key generation, signature generation and verification of our matrix-based Rainbow are described 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 introduced above. Since L and R are affine transformations, $L^{-1}(\mathbf{M})$ and $R^{-1}(\mathbf{S}')$ can be easily computed.
• **Verification** If $F(\mathbf{S}) = \mathbf{M}$, the signature is accepted. Otherwise, it is rejected.

We call this scheme matrix-based Rainbow and it is denoted by M-Rainbow$(K; v_1, d_1 * o'_1, \ldots, d_t * o'_t)$. We call $v_1, d_1, o'_1, \ldots, d_t, o'_t$ a parameter of matrix-based Rainbow. M-Rainbow$(K; v_1, d_1 * o'_1, \ldots, d_t * o'_t)$ is not only a variant of Rainbow$(K; v_1, d_1 o'_1, \ldots, d_t o'_t)$, but also a variant of Rainbow$(K; v_1, \overbrace{o'_1, \ldots, o'_1}^{d_1}, \ldots, \overbrace{o'_t, \ldots, o'_t}^{d_t})$.

4. SECURITY OF MATRIX-BASED RAINBOW

In this section, we analyze the security of our scheme with respect to known attacks against Rainbow. Before discussing the security, let us establish some notations for our scheme. For the public key $F = (f^{(v_1+1)}, \ldots, f^{(n)})$ of M-Rainbow$(K;$
$v_1, d_1 * o'_1, \ldots, d_t * o'_t)$, each quadratic polynomial $f^{(h)}$ is expressed by

$$f^{(h)}(\mathbf{x}) = \mathbf{x}^T \overline{A}^{(h)} \mathbf{x} + \overline{B}^{(h)} \mathbf{x} + \overline{C}^{(h)} \quad (h = v_1 + 1, \ldots, n),$$

where $\overline{A}^{(h)}$ is a square matrix with size n, $\overline{B}^{(k)} \in K^n$ and $\overline{C}^{(k)} \in K$. We denote $\overline{D}^{(h)} = \overline{A}^{(h)} + (\overline{A}^{(h)})^T$ for any h. Similarly, we denote $D^{(h)} = A^{(h)} + (A^{(h)})^T$ where $A^{(h)}$ is a square matrix appearing in (4). We write R_1 for the linear transformation part of the affine transformation R, and Ω_G for the space of linear combinations of $D^{(v_1+1)}, \ldots, D^{(n)}$. Then, for any h, $(R_1^T)^{-1} \overline{D}^{(h)} R_1^{-1}$ is expressed as a linear combination of $D^{(v_1+1)}, \ldots, D^{(n)}$. We denote $\mathcal{O}_t = R_1^{-1}(\{0\}^{n-o_t} \times K^{o_t})$ as the subspace of K^n corresponding to the subspace $\{0\}^{n-o_t} \times K^{o_t}$ in the final layer.

4.1 HighRank Attack

In a HighRank attack [7, 5, 11], one must find a linear combination of $\overline{D}^{(1)}, \ldots, \overline{D}^{(o)}$, whose rank is not full, by conducting an exhaustive search. To estimate the complexity of HighRank attack, we compute the probability of finding such a matrix. It has the same probability as that of finding an element of Ω_G whose rank is not full. If we write

$$(R_1^T)^{-1} \overline{D}^{(h)} R_1^{-1} = \left(\begin{array}{c|c} E & H \\ \hline H^T & 0 \end{array} \right), \tag{6}$$

where E and H are matrices of size $v_t \times v_t$ and $v_t \times o_t$, respectively, H can be expressed as

$$H = \left(\begin{array}{c|c|c} \sum_{j=1}^{o'_t} \lambda_{1j} \mathbf{a}_j^{(t)} & \cdots & \sum_{j=1}^{o'_t} \lambda_{dj} \mathbf{a}_j^{(t)} \end{array} \right). \tag{7}$$

where the λ_{ij}'s are determined by the coefficients of the affine transformation L and we have assumed that the λ_{ij}'s are uniformly distributed.

We can assume that the probability of finding an element of Ω_G whose rank is not full is equal to the probability that H is not full rank because the upper-triangular matrix $A_{00}^{(h)}$ of size $v_t \times v_t$ located at the upper left in $A^{(h)}$ are randomly chosen for any $h = v_t + 1, \ldots, n$. To ensure that H is full rank $(= o_t)$, it is necessary that $o'_t \geq d_t$ $(\Leftrightarrow {o'_t}^2 \geq o_t = o'_t d_t)$ because of the following lemma.

LEMMA 4.1. *H has at most rank ${o'_t}^2$.*

PROOF. Since each block in (7) is spanned by $\mathbf{a}_1^{(t)}, \ldots, \mathbf{a}_{o'_t}^{(t)}$, if we fix $j = 1, \ldots, o'_t$ then the matrix formed from the concatenation of all the $(lo'_t + j)$-th columns for $l = 1, \ldots, o'_t$ is at most rank o'_t. Since the number of js is o'_t, the rank of H is less than or equal to ${o'_t}^2$. □

In what follows, we assume that $o'_t \geq d_t$. We define a matrix Δ of size $v_t \times {o'_t}^2$ by

$$\Delta = \left(\begin{array}{c|c|c|c} \mathbf{a}_1^{(t)} & \mathbf{a}_2^{(t)} & \cdots & \mathbf{a}_{o'_t}^{(t)} \end{array} \right).$$

We suppose that $\mathbf{a}_1^{(t)}, \ldots, \mathbf{a}_{o'_t}^{(t)}$ are chosen such that Δ is full rank. When we set $\Lambda = (\lambda_{ij})$, which is a matrix with size $d_t \times o'_t$, H can be expressed as $H = \Delta \circ (\Lambda \otimes \mathbf{1}_{o'_t})$ where

$$\Lambda \otimes \mathbf{1}_{o'_t} = \left(\begin{array}{c|c|c|c} \lambda_{11}\mathbf{1}_{o'_t} & \lambda_{21}\mathbf{1}_{o'_t} & \cdots & \lambda_{d_t 1}\mathbf{1}_{o'_t} \\ \hline \lambda_{12}\mathbf{1}_{o'_t} & \lambda_{22}\mathbf{1}_{o'_t} & \cdots & \lambda_{d_t 2}\mathbf{1}_{o'_t} \\ \hline \vdots & \vdots & \ddots & \vdots \\ \hline \lambda_{1o'_t}\mathbf{1}_{o'_t} & \lambda_{2o'_t}\mathbf{1}_{o'_t} & \cdots & \lambda_{d_t o'_t}\mathbf{1}_{o'_t} \end{array} \right). \tag{8}$$

PROPOSITION 4.1. *The probability that H is not full rank is less than $(q^{d_t - o'_t} + q^{o_t - v_t})/(q-1)$.*

PROOF. If H does not have full rank, from (8), either $\operatorname{Ker}\Delta \cap \operatorname{Im}(\Lambda \otimes \mathbf{1}_{o'_t}) \neq \{0\}$ or $\operatorname{Rank}(\Lambda \otimes \mathbf{1}_{o'_t}) < o_t$ must hold. Let w_1 be an element in $\operatorname{Im}(\Lambda \otimes \mathbf{1}_{o'_t})$. The probability that w_1 does not belong to $\operatorname{Ker}\Delta$ is roughly equal to $1 - q^{{o'_t}^2 - v_t}/q^{{o'_t}^2} = 1 - q^{-v_t}$ because it is supposed that Δ is full rank. Next, let w_2 be another element in $\operatorname{Im}(\Lambda \otimes \mathbf{1}_{o'_t})$. The probability that w_2 does not belong to the subspace spanned by $\operatorname{Ker}\Delta$ and w_1 is roughly equal to $1 - q^{-v_t+1}$ if w_1 does not belong to $\operatorname{Ker}\Delta$. Continuing this argument, we find that the probability that $\operatorname{Ker}\Delta \cap \operatorname{Im}(\Lambda \otimes \mathbf{1}_{o'_t}) = \{0\}$ is roughly equal to

$$(1 - q^{-v_t})(1 - q^{-v_t+1}) \cdots (1 - q^{-v_t+o_t-1})$$
$$(> 1 - \textstyle\sum_{i=0}^{o_t-1} q^{-v_t+i} > 1 - q^{-v_t+o_t}/(q-1)).$$

Therefore, the probability that $\operatorname{Ker}\Delta \cap \operatorname{Im}\Lambda \otimes \mathbf{1}_{o'_t} \neq \{0\}$ is less than $q^{-v_t+o_t}/(q-1)$. $\operatorname{Rank}(\Lambda \otimes \mathbf{1}_{o'_t}) < o_t = o'_t d_t$ is equivalent to $\operatorname{Rank}(\Lambda) < d_t$. Similarly, the probability that $\operatorname{Rank}(\Lambda) < d_t$ is less than $q^{d_t - o'_t}/(q-1)$. Therefore we have thus proved the proposition. □

COROLLARY 4.1. *If $q > 2$ and $v > o_t + o'_t - d_t$ then the probability that H is not full rank is less than $2q^{d_t - o'_t - 1}$.*

PROPOSITION 4.2. *If $q > 2$ and $v_t > o_t + o'_t - d_t$ then the complexity of HighRank attack against matrix-based Rainbow is $q^{o'_t - d_t + 1} \cdot n^3/12$ **m**.*

Here, **m** denotes the field multiplication. From the above proposition, the complexity of the HighRank attack depends on the parameters o'_t, d_t of the last layer (and n).

4.2 MinRank Attack

In a MinRank attack [7, 15], we must conduct an exhaustive search to find a point $w \in K^n$ such that $w \in \operatorname{Ker} \overline{D}$, where $\overline{D}$ is a matrix with the minimal rank in $\operatorname{Span}\{\overline{D}^{(h)}\}$. To estimate the complexity of a MinRank attack, we will compute the probability of finding such a point. The probability is the same as that of finding $w' \in K^n$ such that $w' \in \operatorname{Ker} D$, where D is a matrix with the minimal rank in Ω_G. Accordingly, the first v_1 components of w' should be zero. Thus, we have

PROPOSITION 4.3. *The probability that for a $w \in K^n$, there is a matrix $\overline{D}$ with the minimal rank in* $\operatorname{Span}\{\overline{D}^{(h)}\}$ *such that $w \in \operatorname{Ker} \overline{D}$ is less than q^{-v_1}.*

PROPOSITION 4.4. *The complexity of a MinRank attack against matrix-based Rainbow is $q^{v_1} \cdot m(n^2/2 - m^2/6)$* **m**.

From the above proposition, the complexity of a MinRank attack depends on the parameters v_1 of the first layer (and m, n).

4.3 Other Attacks

The UOV attack [9, 8, 3], direct attacks [1, 3], UOV-Reconciliation attack [5, 10] and the Rainbow-Band-Separation attack [5, 10] are known attacks against the original Rainbow. The complexity of a UOV attack against the original Rainbow scheme was estimated as follows [8].

UOV attack: $q^{n-2o_t-1} \cdot o_t^4$ **m**.

The security of other attacks are estimated experimentally using MAGMA [2] which contains an efficient implementation of Faugeres $F4$-algorithm [6] for computing Gröbner Basis. As a result, there is no evident difference between the security of our proposed scheme and that of the original Rainbow against these attacks.

Table 1: Results of the experiments with direct attacks over $GF(256)$

$(v_1, d_1 * o'_1, d_2 * o'_2)$	(4,1*3,2*2)	(5,1*3,2*2)	(3,1*4,2*2)
M-Rainbow	5.30 s	11.76 s	13.85 s
Rainbow	5.34 s	11.70 s	13.84 s
random system	5.36 s	11.72 s	13.88 s

Table 2: Results of the experiments with UOV-R attack over $GF(256)$

$(v_1, d_1 * o'_1, d_2 * o'_2)$	(4,4*1,1*5)	(5,5*1,2*2)	(5,5*1,1*5)
M-Rainbow	5.15 s	9.28 s	14.16 s
Rainbow	5.10 s	9.33 s	14.21 s

Table 3: Results of the experiments with RBS attack over $GF(256)$

$(v_1, d_1 * o'_1, d_2 * o'_2)$	(3,1*3,2*2)	(4,1*3,2*2)	(5,1*3,2*2)
M-Rainbow	3.56 s	7.89 s	17.50 s
Rainbow	3.57 s	7.87 s	17.46 s

4.4 Examples and Comparison

We will now give an example of matrix-based Rainbow and compare it with the original Rainbow in terms of secret key size and efficiency at the same security level.

Consider examples with two layers ($\leftrightarrow t = 2$ in the notation of § 3.2). Petzoldt et al. [10] discussed the security of the original Rainbow Rainbow$(K; v_1, o_1, o_2)$ with two layers and the finite field with 256 elements, $K = GF(256)$. In particular, Rainbow$(GF(256); 31, 19, 24)$ has 100-bit against direct attacks, UOV-Reconciliation attack, Rainbow-Band-Separation attack and UOV attack. The example of matrix-based Rainbow presented here is a variant of the original Rainbow, and has the same security as the corresponding Rainbow against not only the above attacks but also Rank attacks.

We defined the notation M-Rainbow$(K; v_1, d_1 * o'_1, d_2 * o'_2)$ for our proposed scheme in the last paragraph of § 3.4. This is a variant of the original Rainbow Rainbow$(K; v_1, o_1, o_2)$ when $o_1 = d_1 * o'_1$ and $o_2 = d_2 * o'_2$. Therefore, K and v_1 are determined by the original Rainbow. We will now determine the values of o'_1, d_1, o'_2, d_2 for matrix-based Rainbow. These values must be determined from the complexities of attacks in § 4, i.e. HighRank, MinRank, direct, UOV, UOV-Reconciliation (UOV-R), and Rainbow-Band-Separation (RBS) attacks. In particular, we imposed the following conditions: (1) $v_1 \geq o_2$, (2) $o'_2 \geq d_2$, and (3) $v_2 \geq o_2 + o'_2 - d_2$. More details on the values of the parameters are given in the example. The secret key sizes and the numbers of multiplications and additions in the signature generation process were compared for M-Rainbow$(K; v_1, d_1 * o'_1, d_2 * o'_2)$ and Rainbow$(K; v_1, o_1, o_2)$. We also experimented and compared the time taken by these schemes to generate a signature. The implementation environment consisted of an Intel Core i5 2.67GHz CPU with 4GB of RAM and a gcc compiler. Tables 5 presents the average timing of 1,000 random instances in our experiment.

4.4.1 Example

This example deals with Rainbow$(k; v_1, o_1, o_2)$ with $k = GF(256)$, $v_1 = 31$, $o_1 = 19$ and $o_2 = 24$, which has 100-bit security. We converted Rainbow$(GF(256); 31, 19, 24)$ into the matrix-based Rainbow: M-Rainbow$(k; v_1, d_1 * o'_1, d_2 * o'_2)$ with the same $k = GF(256)$ and $v_1 = 31$ but with $d_1 = 19, o'_1 = 1$ and $d_2 = 2, o'_2 = 12$. From the above discussion of the matrix-based Rainbow, we estimate the bit security of 264, 103, 218, 100 against MinRank attack (MinRank), HighRank attack (HighRank), UOV attack (UOV) and other attacks (Direct, RBS and UOV-R) (See Table 1). Therefore, the total bit security of M-Rainbow$(GF(256); 31, 19 * 1, 2 * 12)$ had 100-bit security, which is equivalent to Rainbow$(GF(256); 31, 19, 24)$. Next, we estimated the secret key size, and the number of multiplications and additions used in M-Rainbow$(GF(256);$
$31, 19*1, 2*12)$: then compared them with those of Rainbow$(GF(256); 31, 19, 24)$. From the formulas in §4.2 in Table 2 we estimated that the secret key size for M-Rainbow$(GF(256); 31, 19*1, 2*12)$ was 56,674 B, which is about 37.2% shorter than that of Rainbow$(GF(256); 31, 19, 24)$. The number of multiplications and additions used in M-Rainbow$(GF(256); 31, 19 * 1, 2 * 12)$ was fewer by about 40% than that of Rainbow$(GF(256); 31, 19, 24)$. Moreover, our implementation in C indicated that the signature generation of the proposed matrix-based Rainbow is about 34% faster than the

Table 4: Comparison of security levels of matrix-based Rainbow and the original Rainbow

M-Rainbow($GF(256); 31, 19*1, 2*12$)	
Attack	Security level (bits)
MinRank	264
HighRank	103
UOV	218
Direct & RBS & UOV-R [10]	100
Lowest	100

⇕

Rainbow($GF(256); 31, 19, 24$)	
Attack	Security level (bits)
MinRank	272
HighRank	208
UOV	218
Direct & RBS & UOV-R [10]	100
Lowest	100

original one at the same security level.

Table 5: Comparison of secret key sizes and efficiency of signature generations of matrix-based Rainbow and the original Rainbow

Rainbow	Original	Matrix-based
Secret Key Size (bytes)	90226	56674 (62.8%)
Multiplication	98636	58938 (59.8%)
Addition	96829	57131 (59.0%)
Experiment (μs)	694	455 (65.6%)

5. CONCLUSION

We presented a variant of Rainbow, that reduces the size of the secret key and speeds up the signature generation process compared with the original Rainbow. We also analyzed the security of our proposed scheme against previously known attacks such as MinRank attack, HighRank attack, and UOV attack. In addition, we presented an explicit parameter of our proposed matrix-based Rainbow for 100-bit security. Using this explicit parameter, our proposed scheme reduces the size of the secret key by about 40% and speeds up the signature generation process by about the same amount. Our experiment in C using an Intel Core i5 CPU confirmed that the signature generation process of our proposed matrix-based Rainbow for 100-bit security is about 34% faster than that of the corresponding original Rainbow at the same security level.

As a part of future work, we plan to examine secure parameters of our proposed scheme using different base fields (e.g., other extension degrees or odd characteristics). We also plan to consider scenarios involving more higher security levels.

6. ACKNOWLEDGMENTS

This work was partially supported by the Japan Science and Technology Agency (JST) Strategic Japanese-Indian Cooperative Programme for Multidisciplinary Research Fields, which aims to combine Information and Communications Technology with Other Fields. The first author is supported by Grant-in-Aid for Young Scientists (B), Grant number 24740078. The second author would like to thank partial support of NSF China grants: 60973131 and U1135004.

8. REFERENCES

[1] Bernstein, D.J., Buchmann, J. and Dahmen, E., "Post Quantum Cryptography", Springer, Heidelberg 2009.

[2] Bosma, W., Cannon, J. and Playoust, C., "The Magma Algebra System I. The User Language", J. Symbolic Comput., vol. 24(3-4), pp. 235–265, 1997.

[3] Braeken, A., Wolf, C. and Preneel, B., "A Study of the Security of Unbalanced Oil and Vinegar Signature Schemes", CT-RSA'05, Springer LNCS vol. 3375, pp. 29–43, 2005.

[4] Ding, J. and Schmidt, D., "Rainbow, a New Multivariable Polynomial Signature Scheme", ACNS'05, Springer LNCS vol. 3531, pp. 164–175, 2005.

[5] 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.

[6] Faugére, J.C., "A New Efficient Algorithm for Computing Groebner Bases (F4)", Journal of Pure and Applied Algebra, vol. 139, pp. 61–88, 1999.

[7] Goubin, L. and Courtois, N.T., "Cryptanalysis of the TTM Cryptosystem", ASIACRYPT'00, Springer LNCS vol. 1976, pp. 44–57, 2000. CANS'05, Springer LNCS vol. 3810, pp. 211–222, 2005.

[8] Kipnis, A., Patarin, J. and Goubin, L., "Unbalanced Oil and Vinegar Schemes", EUROCRYPT'99, Springer LNCS vol. 1592, pp. 206–222, 1999.

[9] Kipnis, A. and Shamir, A., "Cryptanalysis of the Oil and Vinegar Signature Scheme", CRYPTO'98, Springer LNCS vol. 1462, pp. 257–266, 1998.

[10] Petzoldt, A., Bulygin, S. and Buchmann, J., "Selecting Parameters for the Rainbow Signature Scheme", PQCrypto'10, Springer LNCS vol. 6061, pp. 218–240, 2010.

[11] Petzoldt, A., Bulygin, S. and Buchmann, J., "CyclicRainbow - A multivariate Signature Scheme with a Partially Cyclic Public Key based on Rainbow", INDOCRYPT'10, Springer LNCS vol. 6498, pp. 33–48, 2010.

[12] Petzoldt, A., Thomae, E., Bulygin, S., and Wolf, C., "Small Public Keys and Fast Verification for Multivariate Quadratic Public Key Systems", CHES'11, Springer LNCS vol. 6917, pp. 475–490, 2011.

[13] Thomae, E., "Quo Vadis Quaternion? Cryptanalysis of Rainbow over Non-Commutative Rings", Cryptology ePrint Archive: Report 2012/270, 2012.

[14] Wolf, C. and Preneel, B., "Taxonomy of Public Key Schemes Based on the Problem of Multivariate Quadratic Equations", Cryptology ePrint Archive, Report 2005/077, 64 pages, 2005.

[15] Yang, B.-Y. and Chen, J.-M., "Building Secure Tame like Multivariate Public-Key Cryptosystems: The new TTS", ACISP'05, Springer LNCS vol. 3574, pp. 518–531, 2005.

[16] Yang, B.-Y., Chen, J.-M. and Chen Y.-H., "TTS: High-speed Signatures on a Low-Cost Smart Card", CHES'04, Springer LNCS vol. 3156, pp. 371–385, 2004.

[17] Yasuda, T., Sakurai, K., and Takagi, T, "Reducing the Key Size of Rainbow using Non-commutative Rings", CT-RSA'12, Springer LNCS vol. 7178, pp. 68–83, 2012.