IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received April 23, 2017, accepted June 8, 2017, date of publication June 19, 2017, date of current version July 17, 2017.

Digital Object Identifier 10.1109/ACCESS.2017.2717279

Circulant Rainbow: A New Rainbow Variant With Shorter Private Key and Faster Signature Generation

ZHINIANG PENG AND SHAOHUA TANG, (Member, IEEE)
School of Computer Science and Engineering, South China University of Technology, Guangzhou 510640, China

Corresponding author: Shaohua Tang (shtang@ieee.org)

This work was supported in part by the National Natural Science Foundation of China under Grant 61632013, Grant U1135004, and Grant 61170080, in part by the 973 Program under Grant 2014CB360501, in part by the Guangdong Provincial Natural Science Foundation under Grant 2014A030308006, and in part by the Guangdong Provincial Project of Science and Technology under Grant 2016B090920081.

ABSTRACT Rainbow is one of the most important signature schemes in multivariate public key cryptography. It enjoys a strong security guarantee and is a promising signature scheme in Post-Quantum Cryptography. However, it suffers from large key size. In this paper, we propose Circulant Rainbow with shorter private key and higher signing efficiency. In Circulant Rainbow, we introduce rotating relations into parts of Rainbow private key to speed up the signing procedure and reduce the private key size. We carefully choose security parameters so that our Circulant Rainbow is secure against all known attacks. In our experiment, Circulant Rainbow is about three times faster than original Rainbow and it can reduce the private key size by about 45%. We also make a comparison of Circulant Rainbow with some traditional signature schemes, the results show that Circulant Rainbow is a promising candidate in Post-Quantum Cryptography.

INDEX TERMS MPKC, Rainbow signature scheme, Post-Quantum Cryptography, AVX2.

I. INTRODUCTION

In [1] and [2], Shor proposed some polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. It posed a serious threat to some existing cryptographic schemes such as RSA and ECC. The Post-Quantum Cryptography [3], which is secure against attacks by quantum computers, has become a hot research area. Multivariate public key cryptography (MPKC) is one of the promising candidates for Post-Quantum Cryptography.

Security of MPKC is based on the hardness of solving a set of quadratic multivariate equations over a finite field, which is called MQ problem [4]. The MQ problem is proven to be NP-hard [5], [6], and quantum computers do not appear to have any advantages in solving it. In addition, MPKC is very suitable for resource constrained devices such as WSN nodes and smart cards.

Since the emergence of the first MPKC scheme: MI [7], there have been many MPKC encryption and signature schemes such as HFE [8], ZHFE [9], STS [10], PMI [11], UOV [12], ABC [13], EFC [14] and so on. However, many of them are broken by various attacks such as Differential attack [15], MinRank attack [16]–[18], Highrank attack [19], [20], algebraic key recovery attack [21] and Direct attack [22], [23]. It is clear that secure MPKC schemes are extremely rare.

Rainbow [24] is one of the most important signature schemes in MPKC. It enjoys a strong security guarantee and a fast verification procedure. None of the existing attacks can cause severe security threats to it. However, Rainbow has not been widely used mainly because of its large key size. Therefore, reducing the sizes of private and public keys of Rainbow is an important research direction.

Petzoldt *et al.* [25] proposed Cyclic Rainbow to reduce public key size of Rainbow and improve the verification speed. They inserted some cyclic relations into generation of public key and accelerated the verification using the relations. Several variants of Rainbow using sparse private keys have been proposed to reduce the private key size and improve the signing process, e.g. Enhanced TTS [20] MB Rainbow [26], NT Rainbow [27]. Although Enhanced TTS was broken in [28] because that it lacks some cross-terms of Vinegar variables and Oil variables, the method of reducing private key size of Rainbow using sparse key [26], [27], [29] survived now. However, we find out that the suggested parameters of MB Rainbow and NT Rainbow can be broken by algebraic key recovery attack using good keys.

A. OUR CONTRIBUTIONS

In this paper, we carefully analyze the security of MB Rainbow and NT rainbow and revise their parameters for 80 bits and 100 bits security. We propose a new Rainbow variant called Circulant Rainbow. It provides a new way to reduce the private key size and improve the signing speed of Rainbow. We carefully choose security parameters to make Circulant Rainbow secure against all known attacks. The private key size of Circulant Rainbow is smaller by 45% than that of original Rainbow. To demonstrate the efficiency, we implement our Circulant Rainbow. The results show that Circulant Rainbow is about 3 times faster than original Rainbow and it outperforms many other signature schemes in both signing and verification speed.

II. RAINBOW AND RAINBOW VARIANTS

In this section, we give a description of Rainbow and its variants.

A. BASIC RAINBOW

Ding and Schmidt proposed a signature scheme called Rainbow, which is a generalization of the Oil-Vinegar signature scheme (OV) [30]. The key point of Rainbow is the idea of a multi-layer Oil-Vinegar system.

Let t be the number of layers in Rainbow. Let $v_1, \ldots, v_{t+1}$ be $t+1$ integers such that $0 = v_0 < v_1 < v_2 < \cdots < v_{t+1} = n$. For i=1,...,t, the set of indices of the i-th layer in Rainbow is defined by integers v_i and $o_i = v_{i+1} - v_i$. The number of equations is $m = \sum_{i=1}^{t} o_i$ and number of variables is n. We call (v_1,o_1,...,o_t) a parameter of Rainbow and denote it by Rainbow($K, v_1, o_1, \ldots, o_t$).

Let G=($g_{v_1+1}, \ldots, g_n$) be a map from K^n to K^m where each g_h is a quadratic polynomial of the form:

$$g_h = \mathbf{x}^T A_h \mathbf{x} + \mathbf{b}_h \mathbf{x} + c_h, \mathbf{x} = (x_1, \ldots, x_n)^T.$$

Let $h = v_i + j$ for $i \in [1, \ldots, t]$ and $j \in [1, \ldots, o_i]$. Here, A_{v_i+j} is a square matrix over K with size n expressed by

$$A_{v_i+j} = \begin{bmatrix} VV_{v_i+j} & VO_{v_i+j} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

where VV_{v_i+j} is a randomly chosen square matrix with dimension v_i and VO_{v_i+j} is a random matrix chosen v_i*o_i matrix. $\mathbf{b}_h$ is a vector in K^n taking the form:

$$\mathbf{b}_{v_i+j} = (\mathbf{b}'_{v_i+j}, \overbrace{0, \ldots, 0}^{n - v_{i+1}}),$$

where $\mathbf{b}'_{v_i+j}$ 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 easily computed. For any vector $\mathbf{y}$=($y_1, \ldots, y_m$)$\in K_m$, its preimage can be computed using Algorithm 1.

Here we give a general description of Rainbow.

Algorithm 1 $G^{-1}(\mathbf{y})$

Input: $\mathbf{y} = (y_1, \ldots, y_m) \in K^m$.
Output: $\mathbf{x} = (x_1, \ldots, x_n) \in K^n$.

1: Randomly choose $s_1, \ldots, s_{v_1} \in K$ and let $i = 1$.
2: Substitute $(x_1, \ldots, x_{v_1})$=$(s_1, \ldots, s_{v_1})$ into $g_{v_i+1}, \ldots, g_{v_i+o_i}$ to get a system of linear equations $L\mathbf{x} = \mathbf{u}$ in o_i variables (If the system is not regular, go back to line 1).
3: Solve the system using Gauss Elimination and obtain a solution $(x_{v_i+1}, \ldots, x_{v_i+o_i})$=$(s_{v_i+1}, \ldots, s_{v_i+o_i})$.
4: Let $i = i + 1$. If $i < t + 1$, go back to line 2.
5: **return** $(x_1, \ldots, x_n)$.

1) PRIVATE KEY

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

2) PUBLIC KEY

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

3) SIGNATURE GENERATION

Suppose the document to be signed is $\mathbf{m}$. Then we sign it as follows:

1) Hash it to $\mathbf{w} \in K^n$.
2) Compute $\mathbf{y} = S^{-1}(\mathbf{w})$.
3) Compute $\mathbf{x} = G^{-1}(\mathbf{y})$ using Algorithm 1.
4) Finally compute $\mathbf{s} = R^{-1}(\mathbf{x})$ as signature.

4) SIGNATURE VERIFICATION

The signer sends a document-signature pair $(\mathbf{m}, \mathbf{s})$ to a receiver. The receiver checks the correctness of the signature by checking if $P(\mathbf{s}) = Hash(\mathbf{m})$. If it matches, the signature is valid. Otherwise, the signature is fake.

B. RAINBOW VARIANTS

Petzoldt *et al.* [25] inserted some special sequences into the generation of public key of Rainbow to save memory. Cyclic Rainbow is a special case of this method. It introduces cyclic relations into public key of Rainbow to reduce the public key size. In the meantime, it improves the verification speed by using the cyclic relations.

Several variants of Rainbow using sparse private keys have been proposed to reduce the private key size and improve the signing process. Enhance TTS was proposed by Yang and Chen in 2005 [20]. The overall idea of the scheme is to use several layers of UOV trapdoors and make them as sparse as possible. It can be viewed as a variant of Rainbow. It admits shorter key size and faster signing speed. However, it was broken by a variant of Rainbow-Band-Separation (RBS) attack in [28] because it lacks cross-terms of Vinegar variables and Oil variables.

Yasuda *et al.* [26] proposed MB Rainbow, which divides each layer of Rainbow into smaller blocks by using diagonal matrix representations. The private key size of the MB Rainbow is smaller by 40% than that of original Rainbow and its signing speed is sped up by 40%. Yasuda *et al.* [27] proposed NT Rainbow, which introduces some rotating relations into Vinegar-Vinegar terms of the central map of Rainbow. It can also be combined with MB Rainbow to improve Rainbow even further [29]. However, we find out that MB Rainbow is vulnerable to a variant of RBS attack and suggested parameter sets of NT Rainbow proposed in [26] are not large enough to resist RBS attack [21]. In order to achieve the same security levels, we have to revise the parameters of them.

C. RBS ATTACK AGAINST MB RAINBOW AND NT RAINBOW

For Rainbow(K,v_1,o_1,o_2), the goal of RBS attack is to find the special equivalent key S' and R' such that

$$F = S \circ G \circ R = S' \circ G' \circ R'$$

for a valid trapdoor G', where S' and R' have the special structure shown in Fig. 1.

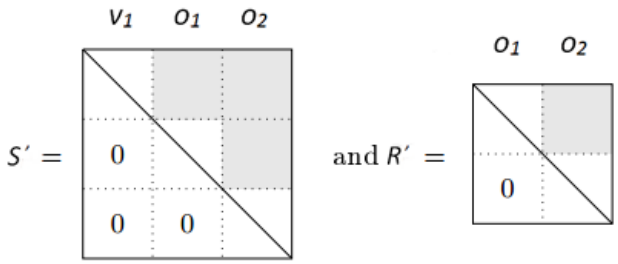


FIGURE 1. Equivalent keys for Rainbow(v_1, o_1, o_2). White parts denote zero elements, gray parts denote arbitrary elements and there are ones at the diagonal.

In RBS attack against Rainbow, there exists good key S'_n and R'_n of the form in Fig. 2.

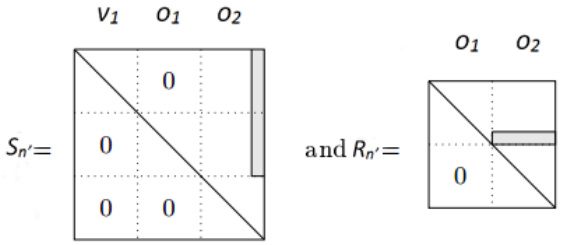


FIGURE 2. Good key for Rainbow(v_1, o_1, o_2).

Only the last column of S'_n contains arbitrary elements in the first two blocks, which are equal to the corresponding values in S'. Respectively, only the second block of the o_1-th row of R'_n contains arbitrary elements, which are equal to the corresponding values in R'. The secret map $(S'^{-1}_n \circ P) \circ R'^{-1}_n$ will have the following form in Fig. 3.

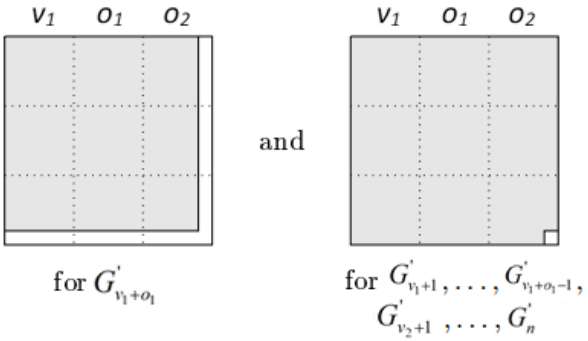


FIGURE 3. Central map of Rainbow(v_1, o_1, o_2) after applying the good key transformation.

Then we can get one cubic equation and $m + n - 2$ quadratic equations in n variables of R'_n and S'_n. To estimate the complexity of solving such a system, we have to calculate the degree of regularity d_{reg} [31], which is the index of the first non-positive coefficient in the Hilbert series $S_{m,n}$ with

$$S_{m,n} = \frac{\prod_{i=1}^{m}(1 - z^{d_i})}{(1 - z)^n},$$

where d_i is the degree of the i-th equation. The computational complexity of solving such a system using F4 algorithm is bounded by

$$O(\binom{n + d_{reg}}{d_{reg}}^{\omega}),$$

where n is the number of variables, m is the number of equations, ω is a linear algebra constant and $2 \leq \omega \leq 3$. In general, we set $\omega = 2$ for cryptanalysis.

For NT Rainbow with suggested parameter (GF(256),18,14,14) for 80 bits security, we can get one cubic equation and 72 quadratic equations in 46 variables when applying RBS attack. The complexity of solving such a system of equations using F4 algorithm is about 2^{70}, which is weaker than the author's claim.

For MB Rainbow with suggested parameter (GF(256),31,19,2*12) for 100 bits security, we can get one cubic equation and 155 quadratic equations in 74 variables when applying RBS attack. The complexity of solving such a system of equations using F4 algorithm is about 2^{110}, which seems to meet the security requirement. However, this attack does not exploit the special key structure of MB Rainbow. In MB Rainbow(K, v_1, o_1, $d * o'_2$), we have more zero columns in the last layer of its central map. After we apply S'^{-1}_n and R'^{-1}_n to the public key P. The map $(S'^{-1}_n \circ P) \circ R'^{-1}_n$ actually have the following form in Fig. 4.

This means that we can get one cubic equation and $(n-1) * (m - o_1 - o'_2 + 1) + (m + 1)$ quadratic equations in n variables of R'_n and S'_n. We give a proof-of-concept code to show this weakness.[1] For MB Rainbow with parameter (GF(256),31,19,2*12), we can actually get one cubic

[1]Proof-of-concept code can be found at https://github.com/edwardz246003/MB-Rainbow

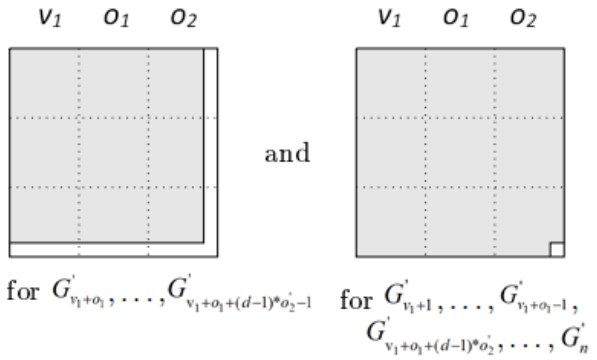


FIGURE 4. Central map of MB Rainbow($K, v_1, o_1, d * o'_2$) after applying the good key transformation.

equation and 993 quadratic equations in 74 variables when applying RBS attack. The complexity of solving such a system of equations using F4 algorithm is about 2^{33} which is much weaker than that in original RBS attack.

1) DISCUSS

MB Rainbow is vulnerable to RBS attack because it has more zero columns in the last layer of its central map. Attackers can get more equations in the first step of RBS attack. To block this attack, we should avoid using MB structure in the last layer of MB Rainbow. Fortunately, other layers of MB Rainbow can still use MB structure to speed up signing process. We can re-select the parameter sets for MB Rainbow to achieve the intended security levels. As for NT Rainbow, the parameter sets proposed in [27] are weaker than the author's claim when applying RBS attack on it. Our revised parameters for MB Rainbow and NT Rainbow will be presented in Section V.

III. DESCRIPTION OF CIRCULANT RAINBOW

In this section, we propose a new Rainbow variant called Circulant Rainbow. Although its name is similar with Cyclic Rainbow, the basic ideas are very different. In Cyclic Rainbow, the authors are able to reduce the public key size and improve the verification speed. But in our Circulant Rainbow, we aim at reducing the private key size and improving the signing speed. Here we start by explaining the basic idea underlying our scheme.

A. BASIC UNDERLYING IDEA

The key idea underlying our scheme is a modification of linear equations appearing in Algorithm 1 of Rainbow signing procedure.

As described in Algorithm 1 of Rainbow signing procedure, we have to solve a system of linear equations described as $L\mathbf{x} = \mathbf{u}$ in the i-th layer of Rainbow. L is a square matrix over K with size o_i, $\mathbf{u}$ is a column vector over K with size o_i and $\mathbf{x}$ is a vector of o_i variables. In general, we use Gauss Elimination to find a solution for $\mathbf{x}$, which takes $O(o_i^3)$ operations on the base field to do this. In Circulant Rainbow, we introduce some rotating relations into parts of Rainbow central map to make L become a circulant matrix [32] which can be inverted efficiently. Here we define a circulant matrix L taking the form:

$$L = \begin{pmatrix} l_1 & l_2 & \cdots & l_{o_i-1} & l_{o_i} \\ l_{o_i} & l_1 & \cdots & l_{o_i-2} & l_{o_i-1} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ l_3 & l_4 & \cdots & l_1 & l_2 \\ l_2 & l_3 & \cdots & l_{o_i} & l_1 \end{pmatrix}.$$

When L is a circulant matrix, we can use the extended Euclidean algorithm to compute the inverse of it. This only takes $O(o_i^2)$ operations on the base field. In addition, the structure introduced in the central map will also improve the speed of the remaining parts of Algorithm 1 significantly.

To make L be a circulant matrix, we introduce some rotating relations into each layer of the central map of Rainbow. Here we first show the matrix representation of the i-th layer central polynomials of Circulant Rainbow. We keep the constant and linear parts so that the central matrices are square matrices with size $v_i + o_i + 1$ of the form in Fig. 5.

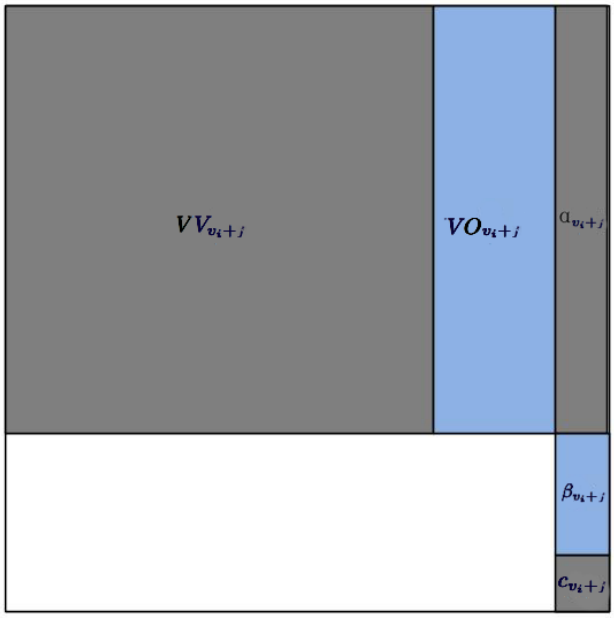


FIGURE 5. Central map matrix of the i-th layer of Circulant Rainbow.

The white areas stand for zero elements. The gray areas stand for arbitrary elements in the base field. The blue areas will have some rotating relations with other central matrices. VV_{v_i+j} is a square matrix with size v_i standing for Vinegar-Vinegar cross-terms coefficients and VO_{v_i+j} is a $v_i * o_j$ matrix standing for Oil-Vinegar cross-terms. α_{v_i+j} in the last column stands for the linear coefficients of Vinegar variables, β_{v_i+j} in the last column stands for the linear coefficients of Oil variables and c_{v_i+j} stands for constant term. Every single central matrix of Circulant Rainbow looks exactly the same as that of original Rainbow.

As we describe above, Circulant Rainbow doesn't have circulant matrix in its central maps. It only has some rotating relations among parts of submatrix of different central matrices. Those rotating relations will help us to get circulant matrices during the signing process. Here we write matrix VO_{v_i+j} in column form. VO_{v_i+j} and β_{v_i+j} have the following rotating relations:

$$\begin{aligned}
VO_{v_i+1} &= (\mathbf{vo}_{i,1}, \mathbf{vo}_{i,2}, \cdots, \mathbf{vo}_{i,o_i}) \\
VO_{v_i+2} &= (\mathbf{vo}_{i,o_i}, \mathbf{vo}_{i,1}, \cdots, \mathbf{vo}_{i,o_i-1}) \\
&\vdots \\
VO_{v_i+o_i} &= (\mathbf{vo}_{i,2}, \mathbf{vo}_{i,3}, \cdots, \mathbf{vo}_{i,1}) \\
\beta_{v_i+1} &= (\beta_{i,1}, \beta_{i,2}, \cdots, \beta_{i,o_i})^T \\
\beta_{v_i+2} &= (\beta_{i,o_i}, \beta_{i,1}, \cdots, \beta_{i,o_i-1})^T \\
&\vdots \\
\beta_{v_i+o_i} &= (\beta_{i,2}, \beta_{i,3}, \cdots, \beta_{i,1})^T.
\end{aligned}$$

B. INVERTING THE CENTRAL MAP

Since our central map G is a special form of the G given in Section II-A, we can use Gauss Elimination to compute the inverse of our G. Here we are going to describe a faster way to invert the central map of Circulant Rainbow.

Assume the vector to be inverted is $\mathbf{y}$ and the Vinegar vector for the i-th layer is $\mathbf{v} = (v_1, \ldots, v_{v_i})$. Substituting $(x_1, \ldots, x_{v_i})$ with $(v_1, \ldots, v_{v_i})$, we will get a linear equation system of o_i variables. For each central polynomial g_{v_i+j} in the i-th layer, we get an equation:

$$\underbrace{\mathbf{v}^T \cdot VV_{v_i+j} \cdot \mathbf{v} + \mathbf{v}^T \cdot \alpha_{v_i+j} + c_{v_i+j}}_{\text{constant}} + \underbrace{\mathbf{v}^T \cdot VO_{v_i+j} \cdot \mathbf{o} + \beta_{v_i+j} \cdot \mathbf{o}}_{\text{linear in } \mathbf{o}} = y_{v_i+j},$$

where vector $\mathbf{o} = (x_{v_i+1}, \ldots, x_{v_i+o_i})$ stands for Oil variables. Let

$$u_{v_i+j} = y_{v_i+j} - (\mathbf{v}^T \cdot VV_{v_i+j} \cdot \mathbf{v} + \mathbf{v}^T \cdot \alpha_{v_i+j} + c_{v_i+j})$$

for $j \in [1, \cdots, o_i]$. Then we get a linear system:

$$\underbrace{\begin{pmatrix} \mathbf{v}^T \cdot VO_{v_i+1} + \beta_{v_i+1} \\ \mathbf{v}^T \cdot VO_{v_i+2} + \beta_{v_i+2} \\ \vdots \\ \mathbf{v}^T \cdot VO_{v_i+o_i-1} + \beta_{v_i+o_i-1} \\ \mathbf{v}^T \cdot VO_{v_i+o_i} + \beta_{v_i+o_i} \end{pmatrix}}_{\mathrm{L}} \begin{pmatrix} x_{v_i+1} \\ x_{v_i+2} \\ \vdots \\ x_{v_i+o_i-1} \\ x_{v_i+o_i} \end{pmatrix} = \begin{pmatrix} u_{v_i+1} \\ u_{v_i+2} \\ \vdots \\ u_{v_i+o_i-1} \\ u_{v_i+o_i} \end{pmatrix}.$$

As matrices VO_{v_i+j} and vector β_{v_i+j} have rotating relations. After plugging in vector $\mathbf{v}$, L will become a circulant matrix with size o_i, which can be inverted efficiently.

1) COMPUTING L

Before talking about how to invert L, we first introduce how to calculate L efficiently. To compute matrix L, we need to compute $\mathbf{v}^T \cdot VO_{v_i+j} + \beta_{v_i+j}$ for $j \in [1, \ldots, o_i]$. Since L is a circulant matrix, we only need to compute the first row of L. The rest of L can be generated by its right rotating sequences. The time complexity of computing L is improved by a factor of o_i in i-th layer of Circulant Rainbow.

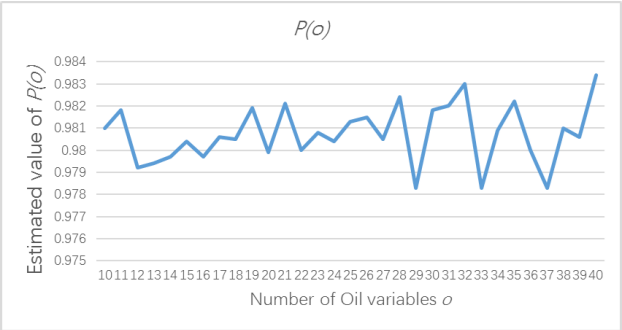


FIGURE 6. Estimated value of $P(o)$.

2) INVERTIBLE PROBABILITY OF L

If the matrix L we get is not invertible, we have to choose another random Vinegar vector $\mathbf{v}$ to get an invertible matrix L. To get a faster signing algorithm, we have to make sure that the invertible probability of a random circulant matrix is large enough. We test the invertible probability $P(o)$ of a random circulant matrix over GF(256) with size o by experiments. We test each $P(o)$ with $o \in [10, 40]$ for 10^5 times and record their average values. Fig. 6 gives the estimated value of $P(o)$.

From Fig. 6, we can observe that $P(o)$ is very close to 1.

3) INVERTING L

In this paper, we use extended Euclidean algorithm to compute the inverse of a circulant matrix L. Suppose L is an invertible circulant matrix over a finite field K with size o. It is well known that the inverse of a circulant matrix is also circulant. We consider the problem of computing a circulant matrix J that $LJ = I$. Let $(l_0, l_1, \ldots, l_{o-1})$ be the first row of L. We can associate L with a polynomial $l(x) = \sum_{k=0}^{o-1} l_i x^i$ over the ring $K[x]$. Computing the inverse of L is equivalent to finding a polynomial $j(x)$ in $K[x]$ such that $l(x) * j(x) = 1 \mod x^o - 1$ [33]. Hence, the problem of inverting a circulant matrix is equivalent to inverting a polynomial in the ring $K[x]/(x^o - 1)$. To find a solution, it takes about $O(o^2)$ arithmetic operations by using the extended Euclidean algorithm which is much faster than Gauss elimination.

C. GENERAL DESCRIPTION OF CIRCULANT RAINBOW

Using the invertible map G, we give a general description of Circulant Rainbow with shorter private key and faster signature generation.

1) KEY GENERATION

According to the required security level, we choose the appropriate set of parameters which is denoted by Rainbow(K,v_1,o_1,…,o_t). Then we randomly generate a quadratic map G according to Section III-B and two affine

transformations S: $K^m \to K^m$ and R: $K^n \to K^n$. The private key consists of (S,G,R) and the public key consists of the composite map $F = S \circ G \circ R : K^n \to K^m$.

2) SIGNATURE GENERATION

Suppose the document to be signed is $\mathbf{m}$. Then we sign it as follows:

1) Hash it to $\mathbf{w} \in K^n$.
2) Compute $\mathbf{y} = S^{-1}(\mathbf{w})$.
3) Compute $\mathbf{x} = G^{-1}(\mathbf{y})$ using the circulant method.
4) Finally compute $\mathbf{s} = R^{-1}(\mathbf{x})$ as signature.

3) SIGNATURE VERIFICATION

The signer sends a document-signature pair ($\mathbf{m}$,$\mathbf{s}$) to a receiver. The receiver checks the correctness of the signature by checking if $P(\mathbf{s}) = Hash(\mathbf{m})$. If it matches, the signature is valid. Otherwise, the signature is fake.

4) CORRECTNESS OF CIRCULANT RAINBOW

Any document-signature pair ($\mathbf{m}$, $\mathbf{s}$) generated by our signing algorithm satisfies the formula $Hash(\mathbf{m}) = S(G(R(\mathbf{s})))$. As the public map P of Circulant Rainbow is a composite map of S, G and R, $P(\mathbf{s}) = Hash(\mathbf{m})$ holds for every document-signature pair generated by Circulant Rainbow.

IV. SECURITY OF CIRCULANT RAINBOW

In this section, we are going to analyze the security of Circulant Rainbow by applying existing attacks on it. As Circulant Rainbow is actually a subset of original Rainbow, all the methods of attacking original Rainbow can also be used to attack Circulant Rainbow. But we will show that Circulant Rainbow is as hard as original Rainbow if we choose the parameters appropriately. All the experiments in this Section are run in MAGMA V2.19 on an computer with a 192GB RAM and an Intel Xeon E5-2660V2 CPU.

A. DIRECT ATTACK

We compare the time taken by the Direct attack against Circulant Rainbow and original Rainbow by experiments. Since the public key of Rainbow is a under-determined system, we fix some of variables before applying an Gröbner basis based algorithm. We carry out a number of experiments with MAGMA [34], which contains an efficient implementation of F4 algorithm [22] to compute Gröbner basis [35], [36] after we fix v_1 variables of Rainbow and Circulant Rainbow. We list the timing results for attacking Circulant Rainbow and original Rainbow by Direct attack in Table 1.

Table 1 shows that there is no significant difference between attack time of original Rainbow and Circulant Rainbow when applying Direct attack. So the lower bound of the complexity of Direct attack using Hybrid F5 algorithm against Circulant Rainbow can be estimated by

$$\min_{k\geq 0} q^k \cdot O(m \cdot \binom{m-k+d_{reg}+1}{d_{reg}}^{\omega}),$$

TABLE 1. Timing results for applying Direct attack on Rainbow and Circulant Rainbow over base field GF(256).

Parameters (v_1,o_1,o_2)	Rainbow	Circulant Rainbow
(4,2,2)	0.01 s	0.01 s
(4,3,3)	0.07 s	0.07 s
(6,4,4)	3.12 s	3.10 s
(6,5,5)	133.73 s	134.41 s
(6,6,6)	5743.53 s	5709.81 s

where the degree of regularity d_{reg} is given as the lowest integer D for which coefficient of z^D in $\frac{(1-z^2)^m}{(1-z)^{m-k}}$ is less than or equal to 0.

B. MinRank ATTACK

MinRank attack is a powerful attack against many MPKC schemes [16]–[18]. The goal of the MinRank attack is to find linear combinations of the central quadratic matrices A_h with minimal rank r, where r is less than or equal to v_2 in Rainbow. Those linear combinations correspond to linear combinations of the central polynomials of the first Rainbow layer. If attackers find these linear combinations, then they will be able to separate the first layer of Rainbow central polynomials. The remaining Rainbow layers can be extracted by a similar technique. The complexity of the attack against original Rainbow can be estimated by $O(o_1 * q^{v_1+1})$.

To estimate the complexity of MinRank attack against Circulant Rainbow, we have to analyze the rank behavior of the central quadratic matrix A_h of Circulant Rainbow. In each A_{v_i+j} for $j \in [1, \ldots, o_i]$, VV_{v_i+j} is a randomly chosen square matrix with dimension v_i. Rank of A_{v_i+j} will be larger than v_i with overwhelming probability. So we can conclude that the complexity of MinRank attack against Circulant Rainbow is also $O(o_1 * q^{v_1+1})$.

C. HighRank ATTACK

In HighRank attack [19], [20], [37], one must find a small kernel shared by a large number of linear combinations of the matrices A_h. For original Rainbow, the complexity of HighRank attack can be estimated by $O(q^{o_t} * \frac{n^3}{6})$, where o_t is the number of Oil variables in the last layer of Rainbow maps.

In Circulant Rainbow, HighRank attack becomes a dangerous attack if parameters are not properly chosen. To estimate the complexity of HighRank attack, we focus on the last layer of Rainbow central maps. We first define a $o_t * o_t$ rotating matrix

$$M = \begin{pmatrix} 0 & 1 & 0 & 0 & 0 \\ 0 & \ddots & \ddots & \ddots & \vdots \\ 0 & \ddots & 0 & 1 & 0 \\ 0 & \ddots & 0 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}.$$

Then we can write

$$R^{-T} A_h R = \begin{bmatrix} E & H \\ H^T & 0 \end{bmatrix},$$

where E and H are matrices of size $v_t * v_t$ and $v_t * o_t$. If E is an invertible matrix, the rank of matrix $R^{-T}A_hR$ can be estimated by $Rank(R^{-T}A_hR) = Rank(E) + Rank(H)$. In Circulant Rainbow, H can be expressed as $H = VO_{v_t+1}\sum_{i=0}^{o_t-1}\lambda_i M^i$, where λ_i are random elements in the base field. As VO_{v_t+1} is a random matrix of size $v_t * o_t$ ($v_t > o_t$), rank of matrix VO_{v_t+1} will equal to o_t with overwhelming probability. Rank of matrix H mainly depends on rank of matrix $\sum_{i=0}^{o_t-1}\lambda_i M^i$. As characteristic polynomial of matrix M is $x^{o_t}-1$, we can compute the rank of matrix $\sum_{i=0}^{o_t-1}\lambda_i M^i$ by

$$Rank(\sum_{i=0}^{o_t-1}\lambda_i M^i) = o_t - degree(gcd(x^{o_t}-1), \sum_{i=o}^{o_t-1}\lambda_i x^i).$$

We can see that the rank of a random linear combination of the matrices A_h mainly depends on the factorization of polynomial $x^{o_t}-1$ over the base field.

Suppose $x^{o_t}-1$ can be factored as $x^{o_t}-1 = \sum_{k=0}^{z} f_k$ over the base field with d_k=degree(f_k) and $d_0 \le d_1 \le \cdots \le d_z$. Algorithm 2 gives the details of HighRank attack to find the space $R^{-1}(O_t)$.

Algorithm 2 HighRank Attack

Input: Parameters (K,v_1,o_1,…,o_t) and public matrices A_1,…,A_m of Rainbow.

Output: A basis of $R^{-1}(O_t)$.

1: Factor $x^{o_t}-1$ as $x^{o_t}-1 = \sum_{k=0}^{z} f_k$ with d_k=degree(f_k) and $d_0 \le d_1 \le \cdots \le d_z$. Let $D = \{d_0, \ldots, d_z\}$ and $B = \{\}$.

2: Get the first element d_k of D and randomly choose λ'_{v_1+1},…,λ'_n. Compute $A_h = \sum_{i=v_1+1}^{n}\lambda'_i A_i$ and find $U = ker(A_h)$.

3: If $dim(U) > 0$, set $(\sum_{i=v_1+1}^{n}\lambda'_i A_i)U = 0$. If the solution set has dimension $o_t - d_k$, add basis of U into B and remove d_k from D. If $D \neq \emptyset$, goto Step 2.

4: **return** B.

The complexity of HighRank attack against Circulant Rainbow can be estimated by

$$HighRank(q, n, o_t) = \frac{n^3}{6}(q^{d_0} + q^{d_1} + \cdots + q^{d_z}),$$

which is determined by the factorization of polynomial $x^{o_t}-1$. For different q and o_t, the factorization of polynomial $x^{o_t}-1$ is very different. For example, the complexity of HighRank attack against Rainbow(GF(256),19,18,19) is about 2^{87}, but the complexity of HighRank attack against Rainbow(GF(256),19,18,18) is only about 2^{41}. In order to prevent HighRank attack, we have to choose q and o_t carefully. Parameters of Circulant Rainbow are less flexible than that of original Rainbow because of HighRank attack. In our experiments, GF(256) appears to be the best choice for the base field of Circulant Rainbow.

D. RAINBOW-BAND-SEPARATION ATTACK

The intrinsic idea of RBS attack is to exploit the sparse key structure of Rainbow [19]. It can be seen as an extension of the UOV-reconciliation attack [19], [38]. Attackers find an equivalent key S' and R' such that $S \circ G \circ R = P = S' \circ G' \circ R'$ for a valid trapdoor G'. It means that S' and R' preserve the structure of G. In the first step of RBS attack, attacker can get one cubic equation and $m+n-2$ quadratic equations in n variables of S' and R' by identifying zero elements in the cross-terms of Vinegar variables and Oil variables in central polynomials. The complexity of RBS attack is mainly determined by solving this system.

In MB Rainbow($K, v_1, o_1, d * o'_2$), there have more zero elements in the cross-terms of Vinegar variables and Oil variables than those of original Rainbow. Attackers can identify more zero elements and get more equations when running RBS attack. This makes MB Rainbow vulnerable to RBS attack. In Circulant Rainbow, we have dense cross-terms just like original Rainbow. Attackers cannot identify more zero elements to run RBS attack in Circulant Rainbow than those in original Rainbow. One may think that the rotating relations might actually give more equations in RBS attack because we have $OV_h[1,1] - OV_{h+1}[1,2] = 0$ in the central map of Circulant Rainbow. This is not right because that the equivalent trapdoor G' in Circulant Rainbow doesn't have the rotating relations anymore for the special S' and T'. The last column of G' of Circulant Rainbow looks exactly like that in original Rainbow. So we can conclude that Circulant Rainbow is as secure as original Rainbow against the RBS attack. The complexity of RBS attack against Circulant Rainbow is about solving a system which consists of one cubic equation and $m+n-2$ quadratic equations in n variables.

E. UOV ATTACK

The goal of UOV attack [39] is to find the preimage of Oil subspace O under the transformation R^{-1}. The complexity of UOV attack can be estimated by $O(q^{v-o-1} * o^4)$ in UOV. In Circulant Rainbow, the transformation S mix all polynomial components of the central map. Therefore, each polynomial of public key can be considered to be a UOV polynomial with v_t Vinegar variables and o_t Oil variables. If we choose $v_t \ge 2o_t$, the complexity of this attack will be exponential.

In order to verify our claim, we generated 100 instances of original Rainbow and Circulant Rainbow and then apply UOV attack against them in MAGMA. Table 2 shows that original Rainbow and Circulant Rainbow almost have the same performance resisting UOV attack. According to the cryptanalysis in [12], we can conclude that the attack complexity of UOV attack against Circulant Rainbow is about $O(q^{v_t-o_t-1} * o_t^4)$.

TABLE 2. Timing results for applying UOV attack on Rainbow and Circulant Rainbow over base field GF(256).

Parameters (v_1,o_1,o_2)	Rainbow	Circulant Rainbow
(4,2,2)	0.03 s	0.03 s
(4,3,3)	0.41 s	0.40 s
(4,4,4)	1.97 s	1.98 s
(5,4,4)	409.41 s	405.96 s
(6,4,5)	947.65 s	943.40 s

F. UOV-RECONCILIATION ATTACK

UOV-Reconciliation (UOV-R) attack was originally designed against UOV. In UOV scheme, the lower right corner of G' must be zero. UOV-R attack exploits this feature to yield o quadratic equations in v variables. As a Rainbow can be viewed as a UOV with v_t Vinegar variables and o_t Oil variables, the UOV-R attack can also be applied to original Rainbow and Circulant Rainbow. As described in Section IV-D, the equivalent trapdoor G' doesn't have the rotating relations for the special S' and R' in UOV-R attack, so attackers can't get more equations in Circulant Rainbow than original Rainbow when applying UOV-R attack. Then we can conclude that the complexity of UOV-R attack against Circulant Rainbow is mainly determined by the complexity of solving o_t quadratic equations in v_t variables.

G. OTHER ATTACKS

One may argue that there may exist some special attacks exploiting the rotating relations of Circulant Rainbow. In fact, rotating relations are hard to exploit in cryptanalysis of MPKC [25], [38], [40]. They are widely used in other areas of cryptography such as lattice-based cryptography [41], [42] and code-based cryptography [43]. From the analysis above, we can conclude that Circulant Rainbow stands against all attacks if we choose the parameters properly.

V. EXPERIMENTS AND COMPARISONS

In this section, we implement Circulant Rainbow and make an overall comparison of Circulant Rainbow with some other signature schemes.

A. COMPARISON WITH OTHER RAINBOW VARIANTS

Here, we compare Circulant Rainbow with other Rainbow variants in terms of efficiency and key size at the same security levels. From the security analysis in Section II-C, we know that MB Rainbow with parameter $(K, v_1, o_1, d * o_2')$ is vulnerable to RBS attack because it has more zero columns in the last layer of its central map. Attackers can identify more equations in the first step of RBS attack. But this does not kill the MB method. In fact, other layers of MB Rainbow can still use the MB structure to speed up signing process. For example, we can use MB Rainbow with parameter $(K, v_1, d * o_1', o_2)$ to block the attack that we mentioned in Section II-C.

Now we estimate the security of Circulant Rainbow and other Rainbow variants against Direct attack, HighRank attack, MinRank attack, RBS attack, UOV attack and UOV-R attack. To further understand the security of Circulant Rainbow and other Rainbow variants, we record the complexities of various attacks against Circulant Rainbow and other Rainbow variants over base field GF(256) in Table 5 and Table 6.

From Table 5 and Table 6, we can observe that Circulant Rainbow is weaker in HighRank attack than original Rainbow. This causes their parameters less flexible. But this has little effect on choosing parameters because the security of Rainbow mainly depends on RBS attack. We can choose (GF(256),19,18,19) and (GF(256),26,23,23) for 80 bits and 100 bits security for Circulant Rainbow, NT Rainbow, and original Rainbow. Then we slightly modify them to fit MB Rainbow.

1) HASH FUNCTION

For digital signatures, we need to have $q^m > 2^l$, where l is the length of the hash, so that $\mathbf{w}$ can hold at least one on hash digest for security consideration. As we have $q^m > 2^{256}$ for 80 bits and 100 bits security for those Rainbow variants, we can choose SHA-256 as hash function for them. However, all the implementations in this paper do not take into account the hash function. We only focus on the basic trapdoor of each scheme.

2) EXPERIMENTAL SETUP

We implement all the Rainbow variants using MAGMA, and record their average performance in signature generation and signature verification. All the schemes are run in MAGMA V2.19 on an Intel Xeon E5-2660V2 CPU.

Table 3 gives a comparison of Circulant Rainbow with other Rainbow variants. From Table 3, we can observe that Circulant Rainbow is about 3 times faster than original Rainbow. It can reduce the private key size by about 45%. Among all the Rainbow variants, Circulant Rainbow is the fastest one in signature generation and has the shortest private key.

B. COMPARISON WITH OTHER SIGNATURE SCHEMES

To further show the efficiency of Circulant Rainbow, we implement our Circulant Rainbow using AVX2 instructions [44] on an Intel Core i7-4790@3.60Ghz CPU. AVX2 is a SIMD instruction set for Intel x64 microprocessors.

TABLE 3. Comparison between Circulant Rainbow and other Rainbow variants.

Security	Scheme	Parameter	PK (Byte)	SK (Byte)	Signature (bit)	Sign (ms)	Ver (ms)
2^{80}	Rainbow	(GF(256),19,18,19)	59.7	42.0	448	46.7	20.0
	MB	(GF(256),19,2*9,19)	59.7	34.2	448	39.6	20.0
	NT	(GF(256),19,18,19)	59.7	26.5	448	25.3	20.0
	Ours	(GF(256),19,18,19)	59.7	23.0	448	15.9	20.0
2^{100}	Rainbow	(GF(256),26,23,23)	121.3	84.7	576	73.2	39.5
	MB	(GF(256),26,2*12,23)	127.5	77.2	584	61.8	41.3
	NT	(GF(256),26,23,23)	121.3	50.8	576	50.7	39.5
	Ours	(GF(256),26,23,23)	121.3	46.1	576	24.8	39.5

TABLE 4. Comparison between Circulant Rainbow and other signature schemes.

Schemes	Parameters	Security (bit)	PK (KB)	SK (KB)	Signature (bit)	Sign (10^3 cycles)	Ver (10^3 cycles)
Gui	(96,5,6,6)	80	61.6	3.1	128	238	62
UOV	(GF(256),56,28)	80	99.0	95.8	656	1685	56
RSA	1024	80	0.13	0.13	1024	475	54
ECDSA	nistk163	80	0.04	0.02	326	720	1440
Ours	(GF(256),18,18,19)	80	59.7	42.0	448	210	33

TABLE 5. Complexities of various attacks against Circulant Rainbow and other Rainbow variants over base field GF(256).

Attacks	Original Rainbow (19,18,19)	MB Rainbow (19,2*9,19)	NT Rainbow (19,18,19)	Circulant Rainbow (19,18,19)
Direct	2^{113}	2^{113}	2^{113}	2^{113}
HighRank	2^{166}	2^{166}	2^{166}	2^{87}
RBS	2^{80}	2^{80}	2^{80}	2^{80}
MinRank	2^{164}	2^{164}	2^{164}	2^{164}
UOV	2^{152}	2^{152}	2^{152}	2^{152}
UOV-R	2^{145}	2^{145}	2^{145}	2^{145}
Security (bit)	80	80	80	80

TABLE 6. Complexities of various attacks against Circulant Rainbow and other Rainbow variants over base field GF(256).

Attacks	Original Rainbow (26,23,23)	MB Rainbow (26,2*12,23)	NT Rainbow (26,23,23)	Circulant Rainbow (26,23,23)
Direct	2^{144}	2^{148}	2^{144}	2^{144}
HighRank	2^{199}	2^{199}	2^{199}	2^{104}
RBS	2^{104}	2^{104}	2^{104}	2^{104}
MinRank	2^{220}	2^{220}	2^{220}	2^{220}
UOV	2^{218}	2^{226}	2^{218}	2^{218}
UOV-R	2^{186}	2^{190}	2^{186}	2^{186}
Security (bit)	100	100	100	100

It expands most integer commands to 256 bits. Traditional asymmetric cryptosystem such as RSA and ECDSA implemented in OpenSSL [45] have already taken the advantage of SIMD instruction. Chen *et al.* [46] gave some SSE implementations of MPKC schemes on modern x64 CPUs, which showed that the advances in chip architecture do not leave MPKC behind while improving traditional alternatives. In this paper, we extend their method into AVX2 to implement Circulant Rainbow.

We compare our Circulant Rainbow implementation with the fastest known Gui, UOV, RSA and ECDSA implementations in terms of the size of the public key, size of the private key, length of the message, length of the signature, signing time and verification time. For Gui, we get the implementation result from [47]. As there is no AVX2 implementation of UOV, we implement it using the same technique as Circulant Rainbow. For RSA and ECDSA, we choose the parameters according to the NIST key management recommendation [48] and record their speeds in the same CPU using OpenSSL speed tester. The overall comparison is given in Table 4.

From Table 4, we can observe that the Circulant Rainbow outperforms all the other signature schemes in terms of signing time and verification time. Compared with other MPKC signature schemes, Circulant Rainbow is better at public key size. But it has larger private key and signature than Gui. Compared with traditional signature schemes, Circulant Rainbow has larger public key and private key. Considering that these keys do not need to be updated frequently in many scenarios, this result is acceptable in the field of MPKC. Anyway, our results show that Circulant Rainbow is comparable to other signature schemes. We believe that it's a promising MPKC signature scheme.

VI. CONCLUSION

In this paper, we proposed a new Rainbow-like multivariate public key digital signature scheme called Circulant Rainbow. Compared with Rainbow and other Rainbow variants, it has faster signature generation and shorter private key. We carefully analyze its security against known attacks such as Direct attacks, HighRank attack, MinRank attack, UOV attack and RBS attack. Our experiments show that Circulant Rainbow is about 3 times faster than original Rainbow and it outperforms many other signature schemes in speed. We believe that it's a promising candidate in Post-Quantum Cryptography.

REFERENCES

[1] P. W. Shor, "Algorithms for quantum computation: discrete logarithms and factoring," in *Proc. 35th Annu. Symp. Found. Comput. Sci.*, Nov. 1994, pp. 124–134.

[2] P. W. Shor, "Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer," *SIAM Rev.*, vol. 41, no. 2, pp. 303–332, 1999.

[3] D. J. Bernstein, J. Buchmann, and E. Dahmen, *Post-Quantum Cryptography*. Berlin, Germany: Springer-Verlag, 2009.

[4] K. Sakumoto, T. Shirai, and H. Hiwatari, "Public-key identification schemes based on multivariate quadratic polynomials," in *Proc. Annu. Cryptol. Conf.*, 2011, pp. 706–723.

[5] M. R. Gary and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. New York, NY, USA: Freeman, 1990.

[6] J. Patarin and L. Goubin, "Trapdoor one-way permutations and multivariate polynomials," in *Proc. Int. Conf. Inf. Commun. Secur.*, 1997, pp. 356–368.

[7] H. Imai and T. Matsumoto, "Algebraic methods for constructing asymmetric cryptosystems," in *Proc. Int. Conf. Appl. Algebra, Algebraic Algorithms, Error-Correcting Codes*, 1985, pp. 108–119.

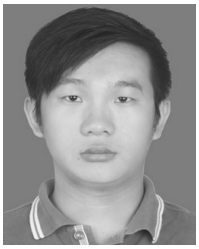
[8] J. Patarin, "Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms," in *Proc. Int. Conf. Theory Appl. Cryptogr. Techn.*, Saragossa, Spain, May 1996, pp. 33–48.

[9] J. Porras, J. Baena, and J. Ding, "ZHFE, a new multivariate public key encryption scheme," in *Proc. 6th Int. Workshop (PQCrypto)*, Waterloo, ON, Canada, Oct. 2014, pp. 229–245.

[10] C. Wolf, A. Braeken, and B. Preneel, "On the security of stepwise triangular systems," *Des., Codes Cryptograph.*, vol. 40, no. 3, pp. 285–302, 2006.

[11] J. Ding, "A new variant of the Matsumoto-Imai cryptosystem through perturbation," in *Proc. Int. Workshop Public Key Cryptograph.*, 2004, pp. 305–318.

[12] A. Kipnis, J. Patarin, and L. Goubin, "Unbalanced oil and vinegar signature schemes," in *Proc. Int. Conf. Theory Appl. Cryptograph. Techn.*, 1999, pp. 206–222.
[13] C. Tao, A. Diene, S. Tang, and J. Ding, "Simple matrix scheme for encryption," in *Proc. PQCrypto*, vol. 13. 2013, pp. 231–242.
[14] A. Szepieniec, J. Ding, and B. Preneel, "Extension field cancellation: A new central trapdoor for multivariate quadratic systems," in *Proc. Int. Workshop Post-Quantum Cryptograph.*, 2016, pp. 182–196.
[15] D. Smith-Tone, "On the differential security of multivariate public key cryptosystems," in *Proc. Int. Workshop Post-Quantum Cryptograph.*, 2011, pp. 130–142.
[16] X. Jiang, J. Ding, and L. Hu, "Kipnis-Shamir attack on HFE revisited," in *Proc. Int. Conf. Inf. Secur. Cryptol.*, 2007, pp. 399–411.
[17] L. Goubin and N. T. Courtois, "Cryptanalysis of the TTM cryptosystem," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2000, pp. 44–57.
[18] J.-C. Faugere, F. Levy-Dit-Vehel, and L. Perret, "Cryptanalysis of MinRank," in *Proc. 28th Annu. Int. Cryptol. Conf.*, Santa Barbara, CA, USA, Aug. 2008, pp. 280–296.
[19] J. Ding, B.-Y. Yang, C.-H. O. Chen, M.-S. Chen, and C.-M. Cheng, "New differential-algebraic attacks and reparametrization of Rainbow," in *Proc. Int. Conf. Appl. Cryptograph. Netw. Secur.*, 2008, pp. 242–257.
[20] B.-Y. Yang and J.-M. Chen, "Building secure tame-like multivariate public-key cryptosystems: The new TTS," in *Proc. Austral. Conf. Inf. Secur. Privacy*, 2005, pp. 518–531.
[21] E. Thomae, "A generalization of the Rainbow Band Separation attack and its applications to multivariate schemes," in *Proc. IACR Cryptol. ePrint Arch.*, 2012, p. 223, 2012.
[22] J.-C. Faugére, "A new efficient algorithm for computing Gröbner bases (F_4)," *J. Pure Appl. Algebra*, vol. 139, no. 1, pp. 61–88, 1999.
[23] M. Bardet, J.-C. Faugere, and B. Salvy, "On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations," in *Proc. Int. Conf. Polynomial Syst. Solving*, 2004, pp. 71–74.
[24] J. Ding and D. Schmidt, "Rainbow, a new multivariable polynomial signature scheme," in *Proc. Int. Conf. Appl. Cryptograph. Netw. Secur.*, 2005, pp. 164–175.
[25] A. Petzoldt, S. Bulygin, and J. Buchmann, "CyclicRainbow—A multivariate signature scheme with a partially cyclic public key," in *Proc. 11th Int. Conf. Cryptol.*, Hyderabad, India, Dec. 2010, pp. 33–48.
[26] T. Yasuda, J. Ding, T. Takagi, and K. Sakurai, "A variant of rainbow with shorter secret key and faster signature generation," in *Proc. 1st ACM Workshop Asia Public-Key Cryptograph.*, 2013, pp. 57–62.
[27] T. Yasuda, T. Takagi, and K. Sakurai, "Efficient variant of rainbow without triangular matrix representation," in *Proc. Inf. Commun. Technol.-EurAsia Conf.*, 2014, pp. 532–541.
[28] E. Thomae and C. Wolf, "Cryptanalysis of enhanced TTS, STS and all its variants, or: Why cross-terms are important," in *Proc. 5th Int. Conf. Cryptol. Africa*, vol. 7374. Ifrane, Morocco, Jul. 2012, pp. 188–202.
[29] T. Yasuda, T. Takagi, and K. Sakurai, "Efficient variant of Rainbow using sparse secret keys," *J. Wireless Mobile Netw., Ubiquitous Comput., Dependable Appl.*, vol. 5, no. 3, pp. 3–13, 2014.
[30] J. Ding, J. E. Gower, and D. S. Schmidt, "Oil-Vinegar signature schemes," in *Multivariate Public Key Cryptosystems*. New York, NY, USA: Springer, 2006, pp. 63–97.
[31] M. Bardet, J.-C. Faugere, B. Salvy, and B.-Y. Yang, "Asymptotic expansion of the degree of regularity for semi-regular systems of equations," in *Proc. Mega*, 2005, pp. 1–14.
[32] I. Kra and S. R. Simanca, "On circulant matrices," *Notices AMS*, vol. 59, no. 3, pp. 368–377, 2012.
[33] D. Bini, G. M. Del Corso, G. Manzini, and L. Margara, "Inversion of circulant matrices over Z_m," in *Proc. 25th Int. Colloq. (ICALP)*, Aalborg, Denmark, Jul. 1998, pp. 719–730.
[34] W. Bosma, J. Cannon, and C. Playoust, "The MAGMA algebra system I: The user language," *J. Symbolic Comput.*, vol. 24, nos. 3–4, pp. 235–265, 1997.
[35] T. Becker and V. Weispfenning, *Gröbner Bases*. New York, NY, USA: Springer, 1993, pp. 187–242.
[36] B. Buchberger, "Gröbner bases," in *Symbolic and Algebraic Computations*. Moscow, Russia: Mir, 1986.
[37] D. Coppersmith, J. Stern, and S. Vaudenay, "Attacks on the birational permutation signature schemes," in *Proc. Annu. Int. Cryptol. Conf.*, 1993, pp. 435–443.
[38] A. Petzoldt, "Selecting and reducing key sizes for multivariate cryptography," Ph.D. dissertation, Dept. Comput. Sci., Technische Univ. Darmstadt, Hessen, Deutschland, 2013.
[39] A. Kipnis and A. Shamir, "Cryptanalysis of the oil and vinegar signature scheme," in *Proc. Annu. Int. Cryptol. Conf.*, 1998, pp. 257–266.
[40] A. Petzoldt and S. Bulygin, "Linear recurring sequences for the UOV key generation revisited," in *Proc. 15th Int. Conf.*, Seoul, South Korea, Nov. 2012, pp. 441–455.
[41] V. Lyubashevsky, C. Peikert, and O. Regev, "On ideal lattices and learning with errors over rings," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.*, 2010, pp. 1–23.
[42] C. Peikert, "How (not) to instantiate ring-LWE," in *Proc. Int. Conf. Secur. Cryptograph. Netw.*, 2016, pp. 411–430.
[43] M. Baldi, *QC-LDPC Code-Based Cryptography*. Springer, 2014.
[44] D. Kanter, "Intel's haswell CPU microarchitecture," Real World Technol., Nov. 2012. [Online]. Available: http://www.realworldtech.com/haswell-cpu/
[45] *Network Security With OpenSSL*. Newton, MA, USA: O'Reilly Media, Jun. 2002. [Online]. Available: https://www.openssl.org/
[46] A. I.-T. Chen *et al.*, "SSE implementation of multivariate PKCs on modern x86 CPUs," in *Proc. 11th Int. Workshop*, Lausanne, Switzerland, Sep. 2009, pp. 33–48.
[47] A. Petzoldt, M.-S. Chen, B.-Y. Yang, C. Tao, and J. Ding, "Design principles for HFEv-based multivariate signature schemes," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.*, 2015, pp. 311–334.
[48] E. Barker, W. Barker, W. Burr, W. Polk, M. Smid, and P. D. Gallagher, "NIST special publication 800–857 recommendation for key management—Part 1: General," Nat. Inst. Standards Technol., Gaithersburg, MD, USA, Tech. Rep. SP 800-57 Pt. 1, Rev. 3, 2012.



ZHINIANG PENG received the B.E. degree from the South China University of Technology in 2013. He is currently pursuing the Ph.D. degree with the School of Computer Science and Engineering, South China University of Technology. His current research interests include lattice-based cryptography, crypto chip design, and multivariate public key cryptography.

SHAOHUA TANG (M'99) received the B.Sc. and M.Sc. degrees in applied mathematics from the South China University of Technology, China, in 1991 and 1994, respectively, and the Ph.D. degree in communication and information system from the South China University of Technology, in 1998. He was a Visiting Scholar with North Carolina State University, USA, and a Visiting Professor with the University of Cincinnati, USA. He has been a Full Professor with the School of Computer Science and Engineering, South China University of Technology, since 2004. He has authored or co-authored over 100 technical papers in journals and conference proceedings. His current research interests include information security, data security, and privacy preserving in cloud computing and big data. He is a member of the IEEE Computer Society.