A New Trapdoor for Constructing Multivariate Signature Schemes: Simple Matrix Signature Scheme

Changze Yin[1,a)] Yacheng Wang[1,b)] Tsuyoshi Takagi[1,c)]

Abstract: Unbalanced Oil and Vinegar signature scheme (UOV), proposed in 1999, is one of the most famous multivariate signature schemes that are secure until now. Simple Matrix Scheme is a multivariate encryption scheme constructed by using matrix multiplication. As a drawback, it has an unneglectable decryption error rate. In our research, we adopt the trapdoor design of UOV and Simple Matrix and extend their ideas to create a new family of signature schemes. This new family utilizes polynomial matrix multiplication to construct a trapdoor, just like in the Simple Matrix. However, unlike conventional multivariate signature schemes, the central map of our new construction can be easily generalized to more complex maps instead of quadratic maps. Moreover, our new construction has a great resistance against existing attacks on multivariate cryptography, and we estimate secure parameters for a simple signature scheme which belongs to our proposed signature family by considering these attacks.

Keywords: Post-Quantum Cryptography, Multivariate Cryptography, UOV, Simple Matrix, Security

1. Introduction

1.1 Research Background

RSA and ECC are known to be long-lived public key cryptosystems and have been put into application for many years. However, Peter Shor[16] proposed efficient algorithms for integer factorization and computing discrete logarithms in 1994, which raised great concern about current cryptographical technologies. The Nation Institute of Standards and Technology (NIST) published a list of candidates[2] for cryptosystems that are resistant against quantum computers (Post-Quantum Cryptography), which drew a great deal of attention in the field of cryptography. Therefore, it is urgent to explore more directions to prevent attacks on quantum computers. Among many candidates for Post-Quantum Cryptography, Multivariate Public Key Cryptosystem (MPKC) is considered to be a good candidate and it has great potential to build cryptosystems for future use.

MPKC uses a set of multivariate quadratic polynomials as its public key, and its security comes from the hardness of solving the multivariate quadratic (MQ)[19] problem, which is proved to be NP-complete. Since the first MPKC, Matsumoto-Imai (MI)[13] scheme was proposed, many researchers have been trying to find more ways for constructing random-like multivariate quadratic polynomials to use on MPKC. So far, there have been proposed many multivariate encryption schemes such as HFE[14], Simple Matrix scheme[18], SRP[20], EFC[17], HFERP[11], and multivariate signature schemes such as UOV[12], SFLASH[15], HFEv-[14], Rainbow[6]. However, many of them could not withstand cryptanalysis such as algebraic attack, linearization attack, differential attack, rank attack, etc.

Among those signature schemes, UOV signature scheme[12] is still proven to be secure under many cryptoanalysis. While UOV requires large public key size to maintain its security, it has few application scenes for practical use. On the other hand, Simple Matrix encryption scheme has an ingenious design of structure of public key, but the decryption failure limits its practicality. To overcome the problems in UOV signature scheme, researchers have brought many thoughts such as a MPKC signature scheme based on block matrices multiplication.

1.2 Our Contribution

In our research, we extract the essential idea of construction of trapdoor in UOV and apply it to polynomial matrix multiplication which is similar to the structure of Simple Matrix signature scheme. Such improved multivariate signature scheme is more flexible and shows great resistance of current attacks. More precisely, we especially compute the complexity of direct attack[3] and minrank attack[10] algorithms. From experiment results, the computation complexity are respectively $\mathcal{O}\left(\binom{3s^2}{s^2+1}^{\omega}\right)$ and $\mathcal{O}(q^{4s-2}s^6)$ which will be discussed in Section 5.

[1] Department of Mathematical Informatics, University of Tokyo
[a)] changze_yin@mist.i.u-tokyo.ac.jp
[b)] yacheng_wang@mist.i.u-tokyo.ac.jp
[c)] takagi@mist.i.u-tokyo.ac.jp

Meanwhile, this new signature scheme utilizes the structure of polynomial matrix multiplication without concern of decryption failure and we call it Simple Matrix signature scheme. In practical implementation, this Simple Matrix signature scheme[18] has small public key and signature size. For example, to achieve128-bit security level, the public key only costs around 31kB of storage. In fact, even we change the central map enumerated in Section 3, the public key size will not change a lot which means this signature scheme could be applied into different situations.

This paper is organized as follows. In section 2, we will provide some backgrounds of Multivariate Public Key Cryptography and introduce some typical schemes. In section 3, we will focus on the construction of our new signature scheme. In section 4, we pick some currently popular attacks and make an analysis of security of this new signature scheme.

2. Preliminaries

In this section, we will provide some basic concepts and constructions of multivariate public key cryptography. Besides, as representatives, unbalanced oil and vinegar signature scheme and simple matrix encryption scheme will be described in the following.

2.1 Multivariate Public Key Cryptography (MPKC)

Based on the hardness of solving MQ problem[19], MPKC includes many splendid schemes using various trapdoors. The basic structure of a MPKC scheme will be described in this section.

We start our construction from a finite field with q elements denoted by $\mathbb{F}_q$. Let n, m be two positive integers and $\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{F}_q^n$ be a vector with n variables. The function of public key in MPKC generally can be written as the composites of three maps:

$$P(\mathbf{x}) = T \circ F \circ S(\mathbf{x}).$$

The maps $T : \mathbb{F}_q^m \to \mathbb{F}_q^m$ and $S : \mathbb{F}_q^n \to \mathbb{F}_q^n$ are invertible linear functions while the map $F : \mathbb{F}_q^n \to \mathbb{F}_q^m$ is a quadratic polynomial function. Especially, we name the map T as the outer affine transformation and the map S as the inner affine transformation. Moreover, the pre-image of map F usually can be easily solved out and we call it the central map in MPKC.

After previous construction, the public key of MPKC consists two parts: the multiplication and addition in finite field $\mathbb{F}_q$ and the quadratic map P. As an user, the secret key consists three maps: T, F, S. To deliver a message or sign a document, the encryption scheme and signature scheme of MPKC are described as follow:

Encryption scheme:

Encrypt: Given a message $\mathbf{m} \in \mathbb{F}_q^n$, compute $\mathbf{c} = P(\mathbf{m})$ as the corresponding cipher text.

Decrypt: For a cipher text $\mathbf{c} \in \mathbb{F}_q^m$, compute $\mathbf{m}\prime = S^{-1}(F^{-1}(T^{-1}(\mathbf{m})))$ as a decryption result.

Signature scheme:

Sign: To sign a document $\mathbf{m}$, we perform three steps:

- Step 1: Compute $\mathbf{y} = T^{-1}(\mathbf{m})$
- Step 2: Find out a solution $\mathbf{x}$ of quadratic equation $F(\mathbf{x}) = \mathbf{y}$
- Step 3: Compute $\mathbf{s} = S^{-1}(\mathbf{x})$ as a signature of $\mathbf{m}$

Verify: To verify the signature $\mathbf{s}$ whether matches the message $\mathbf{m}$, check the correctness of equation $\mathbf{s} = P(\mathbf{m})$.

2.2 Unbalanced Oil and Vinegar signature scheme(UOV)[12]

As a representative of MPKC signature scheme, UOV remains secure after many attacks. However, the public key size of UOV is quite large which is a concern for practical use. The construction of UOV defines as follows.

Let o, v be positive integers and $n = o+v$. In UOV signature scheme, the first v variables are called vinegar variables and the rest part are called oil variables. The central map in UOV could be written as

$$F = (f_1, \ldots, f_o)$$

where each function $f_l, l \in \{1, \ldots, o\}$ is a quadratic polynomial with the following construction:

$$f_l = \sum_{i=1}^{v}\sum_{j=1}^{v} a_{ij}^l x_i x_j + \sum_{i=1}^{v}\sum_{j=1}^{o} b_{ij}^l x_i x_{j+v} + \alpha(x_1, \ldots, x_n).$$

and $\alpha(x_1, \ldots, x_n)$ is a linear function of $(x_1, \ldots, x_n)$. Obviously each function in f_l does not contain quadratic terms of oil variables which implies the central map is a linear function of oil variables if vinegar variables are fixed. The public key of UOV is a composition of central map F and inner affine transformation S namely

$$P = F \circ S.$$

In this case, adding outer affine transformation or not does not effect its security level. Therefore, the signature generation process can be reduced into following two steps:

(1) Randomly choose values of vinegar variables denoted by $(\tilde{x}_1, \ldots, \tilde{x}_v)$. For a given message $\mathbf{m}$, solve linear equation group

$$\mathbf{m} = F(\tilde{x}_1, \ldots, \tilde{x}_v, x_{v+1}, \ldots, x_n).$$

Mark the solution as $(\tilde{x}_{v+1}, \ldots, \tilde{x}_n)$.

(2) Compute $\mathbf{s} = S^{-1}(\tilde{x}_1, \ldots, \tilde{x}_n)$ as a signature of message $\mathbf{m}$.

2.3 Simple Matrix encryption scheme(ABC)[18]

ABC encryption is a MPKC encryption scheme using polynomial matrix multiplications. Until now, there are few attacks could break ABC encryption scheme, but the decryption failure obstructs practical applications which is

unneglectable especially using small parameters.

In ABC encryption scheme, we define a positive integer s and let $n = s^2, m = 2s^2$. In order to construct a central map, there are three matrices needed to be claimed. Let $s \times s$ square matrix A be:

$$A = \begin{pmatrix} x_1 & \cdots & x_s \\ \vdots & \ddots & \vdots \\ x_{s^2-s+1} & \cdots & x_{s^2} \end{pmatrix}$$

in which $(x_1, \ldots x_n) \in \mathbb{F}_q^n$ is a plaintext vector. Similarly, matrices B and C are in $s \times s$ size:

$$B = \begin{pmatrix} b_1 & \cdots & b_s \\ \vdots & \ddots & \vdots \\ b_{s^2-s+1} & \cdots & b_{s^2} \end{pmatrix}$$

$$C = \begin{pmatrix} c_1 & \cdots & c_s \\ \vdots & \ddots & \vdots \\ c_{s^2-s+1} & \cdots & c_{s^2} \end{pmatrix}$$

Each element in B and C is a linear combination of variables $(x_1, \ldots, x_n)$ with random coefficients. By multiplying matrix A with B and C respectively, we can obtain two quadratic polynomial matrices denoted by E_1 and E_2 which are

$$E_1 = AB \qquad E_2 = AC.$$

Let $f_{(i-1)s+j}$ and $f_{s^2+(i-1)s+j}$ be respectively the (i,j)-th coordinate element in E_1 and E_2. We rearrange E_1 and E_2 into sequence form as central map:

$$F = (f_1, \ldots, f_{s^2}, f_{s^2+1}, \ldots, f_{2s^2}).$$

Hence, the public key of ABC contains the finite field $\mathbb{F}_q$ and a quadratic map:

$$P = T \circ F \circ S.$$

the secret key is a 4-tuple (T, S, B, C). To encrypt a message $\mathbf{m} \in \mathbb{F}_q^n$, we directly compute $\mathbf{c} = P(\mathbf{m})$ as the corresponding cipher text $\mathbf{c}$. Decrypting such cipher text $\mathbf{c}$ contains the following steps:

Step 1: Compute $\mathbf{y} = T^{-1}(\mathbf{c})$ and rewrite the vector $\mathbf{y}$ into matrices form:

$$\bar{E}_1 = \begin{pmatrix} \bar{y}_1 & \cdots & \bar{y}_s \\ \vdots & \ddots & \vdots \\ \bar{y}_{s^2-s+1} & \cdots & \bar{y}_{s^2} \end{pmatrix},$$

$$\bar{E}_2 = \begin{pmatrix} \bar{y}_{s^2+1} & \cdots & \bar{y}_{s^2+s} \\ \vdots & \ddots & \vdots \\ \bar{y}_{2s^2-s+1} & \cdots & \bar{y}_{2s^2} \end{pmatrix}$$

Step 2: To find the solution of $F(\mathbf{x}) = \mathbf{y}$, we have several situations:

- **Case 1**: If $\bar{E}_1$ or $\bar{E}_2$ is invertible, we can find

$$B\bar{E}_1^{-1}\bar{E}_2 = C \quad \text{or} \quad C\bar{E}_2^{-1}\bar{E}_1 = B$$

from the relation in central map. By solving this linear system with n variables and n equations, we can find an unique solution defined as $\bar{x}$.

- **Case 2**: If $\bar{E}_1$ and $\bar{E}_2$ are not invertible but A is invertible, we can view A^{-1} as an unknown matrix noted as W. Thus the central map yields:

$$A^{-1}\bar{E}_1 = W\bar{E}_1 = B$$
$$A^{-1}\bar{E}_2 = W\bar{E}_2 = C$$

which is a linear system with m variables and m equations.

- **Case 3**: If A is a singular matrix, the decryption process causes a failure.

Step 3: If we can find a solution denoted by $\bar{x}$ in step 2, compute $\mathbf{m} = S^{-1}(\bar{x})$.

2.4 Block Matrix Multiplication signature scheme[4]

Here's an another signature scheme needed to be mentioned which is proposed by using polynomial matrix multiplication. Though the idea of designing the trapdoor in Block Matrix Multiplication signature scheme is novel, the public key size is large and the signature generation can be failed with high possibilities.

Before the construction of public key, we introduce the following lemma:

Lemma 1 Let $\mathbb{F}_q$ be a finite field with q elements and $u, v, s = u + v$ be positive integers. We define several matrices in the following. $A \in \mathbb{F}_q^{u\times u}$ and $E \in \mathbb{F}_q^{v\times v}$ are invertible constant matrices. Without loss of generality, we set matrix $B \in \mathbb{F}_q[x_1, \ldots, x_n]^{u\times v}$ as a linear polynomial matrix. $C \in \mathbb{F}_q^{v\times u}$ is a constant matrix. Suppose

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix},\ D = CA^{-1}B + E.$$

Then the matrix M is invertible and

$$M^{-1} = \begin{pmatrix} I & -A^{-1}B \\ O & I \end{pmatrix} \begin{pmatrix} A^{-1} & O \\ O & E^{-1} \end{pmatrix} \begin{pmatrix} I & O \\ -CA^{-1} & I \end{pmatrix}$$

the public key of Block Matrix Multiplication signature scheme is constructed as follows. Let $\mathbb{F}_q$ be a finite field and positive integers $u, v, s = u + v$. The number of unknowns is n and the number of equations is $m = s^2$. Define a matrix $P \in \mathbb{F}_q[x_1, \ldots, x_n]^{s\times s}$:

$$P = \begin{pmatrix} p_1p_1' & \cdots & p_sp_s' \\ \vdots & \ddots & \vdots \\ p_{s^2-s+1}p_{s^2-s+1}' & \cdots & p_{s^2}p_{s^2}' \end{pmatrix}$$

in which $p_i, p_i' \in \mathbb{F}_q[x_1, \ldots, x_n]$ are linear functions and thus P is a quadratic polynomial matrix. The central map is a permutation of matrix H:

$$H = MP = \begin{pmatrix} f_{11} & \cdots & f_{1s} \\ \vdots & \ddots & \vdots \\ f_{s1} & \cdots & f_{ss} \end{pmatrix}$$

where matrix M is defined before. Then we enumerate (f_{ij}) as sequence form $(f_1, \ldots, f_m)$ and the public key can be written as:

$$P = T \circ F \circ S,\ F = (f_1, \ldots, f_m)$$

where T, S are linear transformations.

To generate a signature for given message $\mathbf{m} \in \mathbb{F}_q^m$, we have to take the following steps:

Step 1: Compute $\mathbf{y} = T^{-1}(\mathbf{m})$ and rewrite $\mathbf{y}$ into matrix form H'.

Step 2: Let polynomials (p_i') be random values (a_i), solve out the solution of equation group:

$$p_i'(x_1, \ldots, x_n) = a_i,\ i \in 1, \ldots, m$$

and examine the equation $M^{-1}H' = P$ whether holds. If the solution does not exist, then repeat step 2 from beginning.

Step 3: Assume we get a solution from step 2 saying $\mathbf{x}$, compute $\mathbf{s} = S^{-1}(\mathbf{x})$ as a signature.

3. A New Family of Signature Scheme

In this section, we will discuss our new signature scheme using polynomial matrix multiplication. Different from others, our signature scheme is not limited by quadratic polynomials though we use quadratic maps as a representative. Also, in some degree, this new scheme can be seen as an extension of UOV.

3.1 Trapdoor Design

As we review the structure of UOV signature scheme, the trapdoor of its central map is based on the lack of quadratic terms of oil variables, and then we can solve a linear system of oil variables by randomly choosing the values of vinegar variables. In other words, the values of oil variables depend on the chosen of vinegar variables. Through this relation, the solution space of equation group $F(\mathbf{x}) = \mathbf{m}$ is essentially equivalent to a feasible space of vinegar variables. So we can expand this idea in common situation. By partitioning the variable space into two parts, one part is set to be feasible variables and the other part is set to be bounded variables. After we assign some values to feasible variables, the complex equation group will remains a solvable system. Using this trick, we can create a new kind of signature scheme.

Let s be a positive integer, $n = 2s^2$ and $m = s^2$. The message vector denotes by $(x_1, \ldots, x_n) \in \mathbb{F}_q^n$ and $\mathbb{F}_q$ is a finite field with q elements. Here we define three matrices as

$$A = \begin{pmatrix} a_{11} & \cdots & a_{1s} \\ \vdots & \ddots & \vdots \\ a_{s1} & \cdots & a_{ss} \end{pmatrix},$$

$$B = \begin{pmatrix} b_{11} & \cdots & b_{1s} \\ \vdots & \ddots & \vdots \\ b_{s1} & \cdots & b_{ss} \end{pmatrix},$$

$$C = \begin{pmatrix} c_{11} & \cdots & c_{1s} \\ \vdots & \ddots & \vdots \\ c_{s1} & \cdots & c_{ss} \end{pmatrix}$$

in which a_{ij}, b_{ij}, c_{ij} are linear combinations of variables $(x_1, \ldots, x_n)$. One of central map can be defined as

$$\bar{F}(x_1, \ldots, x_n) = AB + BC.$$

Obviously, such function is a $s \times s$ matrix of quadratic polynomials. In order to composite with outer transformation, we stretch matrix $\bar{F}$ into a sequence form:

$$F = (f_{11}, \ldots, f_{1s}, f_{21}, \ldots, f_{2s}, \ldots, f_{ss}).$$

In final step, we apply the general MPKC signature structure to generate the public key:

$$P = T \circ F \circ S.$$

Naturally, the question is that how we use this structure to produce a signature with given message $\mathbf{m} \in \mathbb{F}_q^m$. Despite multiplying the reverse of two affine transformation, we focus on how to figure out the solution of equation system $F(x_1, \ldots, x_n) = \mathbf{y}, \mathbf{y} \in \mathbb{F}_q^m$. We first rearrange vector $\mathbf{y}$ into matrix form with lexicographic order marked as Y. The equation system turns into solving

$$AB + BC = Y.$$

We then randomly choose a constant matrix $D \in \mathbb{F}_q^{s \times s}$. After we replace matrix B into D, the rest part will be transferred into a linear system:

$$\begin{aligned} B(x_1, \ldots, x_n) &= D \\ A(x_1, \ldots, x_n)D + DC(x_1, \ldots, x_n) &= Y \end{aligned}$$

This linear system consists n variables and n equations. By choosing a proper constant matrix D, we can easily solve out a solution and denote by $\mathbf{x}$.

Another interesting thing is that when we choose a special $B(x_1, \ldots, x_n)$, for example, we choose

$$B(x_1, \ldots, x_n) = B'(x_1, \ldots, x_m)$$

which means matrix B is a function of first m variables, $(x_1, \ldots, x_m)$ can be directly solved out from equation $B(x) = D$. In this case, the central map is retreated into balanced oil and vinegar scheme. To make it "unbalanced", we can just change the size of A, B, C into rectangle matrices.

Actually, in previous discussion, we choose function $AB + BC$ as one of this new family of signature scheme. There are still many alternative choices and we provide several examples in the following:

$$\bar{F}(x) = A(x)B(x) + C(x)$$
$$\bar{F}(x) = A(x)B(x) + B(x)C(x) + B(x)D(x)B(x) + E(x)$$
$$\bar{F}(x) = A(x)B(x) + \phi(B(x))$$

In the first example, we erase the matrix $B(x)$ before matrix $C(x)$, or we add several terms like the structure of 'ideal generated by matrix $B(x)$' in the second one. The key point is that when we fix the matrix $B(x) = D$, the rest part of formula remains a linear system and we can combine equation $B(x) = D$ to create a solvable linear system. Based on this thought, we can even add a function not only quadratic polynomial function but also a much more complicated function $\phi(B(x))$ such as exponential function.

3.2 Description of Our New Scheme

We conclude our idea and create the following signature generating process:

Public Key:

- Finite field $\mathbb{F}_q$ with addition and multiplication
- A set of quadratic polynomial P with:

$$P = T \circ F \circ S, \quad F(x_1, \ldots, x_n) = AB + BC$$

Secret Key:

- Affine transformations T, S.
- Coefficients in matrices A, B, C

Signature generation: Given a message $\mathbf{m}$

- Step 1: Compute $\mathbf{y} = T^{-1}(\mathbf{m})$ and rearrange it into matrix form Y
- Step 2: Randomly choose a constant matrix $D \in \mathbb{F}_q^{s \times s}$, solve the following linear system:

$$B(x_1, \ldots, x_n) = D$$
$$A(x_1, \ldots, x_n)D + DC(x_1, \ldots, x_n) = Y$$

  If the coefficient matrix of unknowns $(x_1, \ldots, x_n)$ is not full rank, repeat step (2) to find a proper constant matrix D. The solution denotes by $\mathbf{x}$.
- Step 3: Compute $\mathbf{s} = S^{-1}(\mathbf{x})$ as the a signature of $\mathbf{m}$.

Verification:

Given a message $\mathbf{m}$ and a signature $\mathbf{s}$, substitute $\mathbf{m}$ and $\mathbf{s}$ into function $P(\mathbf{s}) = \mathbf{m}$, then returns TRUE if this formula holds or FALSE if not.

4. Security Analysis

In this section, we run several attack algorithms to estimate the security level of our new family of signature scheme. Additionally, we will provide the computation complexity of those attacks onto our new scheme.

4.1 Direct Attack

Considering a quadratic equation system:

$$\begin{cases} f_1(x_1, \ldots, x_n) = 0 \\ f_2(x_1, \ldots, x_n) = 0 \\ \quad \cdots \\ f_m(x_1, \ldots, x_n) = 0 \end{cases}$$

To solve out this equation group, we have Linearization method such as XL algorithm[3] and Gröebner Basis method such as F_4[8] or F_5[9] algorithms. No matter what algorithm is, the basic idea of direct attack is to generate several equations to make the whole system become solvable. In our experiment, we chose F_4 algorithm to test the resistance of direct attack. The computation complexity of F_4 is bounded by

$$\mathcal{O}\left(\binom{n + d_{reg} - 1}{d_{reg}}^{\omega}\right)$$

where n is the number of unknowns, d_{reg} is a parameter called the degree of regularity (d_{reg}) which is the minimum degree of generating a solvable system, $2 \leq \omega \leq 3$ is a constant and we pick $\omega = 2.4$ for trivial case. From the experiment result, the degree of regularity of applying F_4 algorithm to our scheme is $s^2 + 1$. Thus, the total complexity in this case is

$$\mathcal{O}\left(\binom{3s^2}{s^2 + 1}^{\omega}\right).$$

Table 1 Experiment results of using F_4 algorithm with different parameters

(q, s, n, m)	d_{reg}	Time(s)
(256,2,8,4)	5	0.009
(256,3,18,9)	10	0.240

4.2 Minrank Attack[10]

Rank attack is one of basic analysis technique which is focus on the low rank property in central map. In our analysis, we select minrank attack to make rank attack. Minrank attack comes from a problem called minrank problem and here is the definition:

Definition 1 Let $\mathbb{F}_q$ be a finite field with q elements, r, n, m are positive integers. For given $m + 1$ matrices $\{M_0, M_1, \ldots, M_m : M_i \in \mathbb{F}_q^{n \times n}\}$, minrank problem is to find a vector $(x_1, \ldots, x_m) \in \mathbb{F}_q^m$ s.t.

$$0 < \text{Rank}(M_0 + \sum_{i=1}^{m} x_i M_i) \leq r.$$

There are many algorithms to solve this problem and we used linear algebra search to perform minrank attack. The basic idea of this method is to find a solution of set

$$\text{Sol} = \{\mathbf{v} = (v_1, \ldots, v_n) \in \mathbb{F}_q^n | \\ (\sum_{i=1}^{m} x_i M_i + M_0)\mathbf{v} = 0 \text{ has non-trivial solution}\}$$

The complexity of minrank attack is bounded by

$$\mathcal{O}(q^{\lceil \frac{m}{n} \rceil r} m^3)$$

and the parameter r is related to the rank of central map F. In our new scheme, the central map can be expressed as

$$\bar{F}(x) = A(x)B(x) + B(x)C(x).$$

More precisely, we write down the matrix form of each component:

$$\begin{pmatrix} f_1 & \cdots & f_s \\ \vdots & \ddots & \vdots \\ f_{s^2-s+1} & \cdots & f_{s^2} \end{pmatrix}$$
$$= \begin{pmatrix} a_{11} & \cdots & a_{1s} \\ \vdots & \ddots & \vdots \\ a_{s1} & \cdots & a_{ss} \end{pmatrix} \begin{pmatrix} b_{11} & \cdots & b_{1s} \\ \vdots & \ddots & \vdots \\ b_{s1} & \cdots & b_{ss} \end{pmatrix}$$
$$+ \begin{pmatrix} b_{11} & \cdots & b_{1s} \\ \vdots & \ddots & \vdots \\ b_{s1} & \cdots & b_{ss} \end{pmatrix} \begin{pmatrix} c_{11} & \cdots & c_{1s} \\ \vdots & \ddots & \vdots \\ c_{s1} & \cdots & c_{ss} \end{pmatrix}$$

The $(i,j)-$th coordinate can be expressed as

$$f_{(i-1)s+j} = \sum_{l=1}^{s} a_{il}b_{lj} + \sum_{l=1}^{s} b_{il}c_{lj}.$$

The rank of matrix form of multiplication $a_{il}b_{lj}$ is 2 and this formula contains $2s-1$ independent terms, therefore the rank of each quadratic map f_i is $4s-2$. Moreover, the complexity of minrank attack to our new scheme is $\mathcal{O}(q^{4s-2}s^6)$.

4.3 Other Attacks

Despite direct attack and minrank attack, we also tried some other attacks such as UOV reconciliation attack[7] and High Order Linearization Equation(HOLE) attack[5]. Since the structure of our new scheme,

$$F(x) = AB + BC$$

the quadratic terms of central map F doesn't have special structure like UOV. Even we know the outer affine transformation T, it's hard to get matrices A, B, C from F because for each coordinate (i,j), the quadratic function $f_{(i-1)s+j}$ consists $2s$ terms of multiplication of linear functions. Moreover, HOLE attack can't work so well because from construction of our new scheme, we cannot find a formula only related to signature and message pairs to get informations from secret key. Therefore, common attacks against ABC and UOV can't work efficiently on our new scheme.

5. Parameters and Implementation

In this section, we focus on the security analysis to formulate parameters for different security levels. Then we will give a sample of computing the public key size and give out a table of our experiment results.

5.1 Parameters

We estimated secure parameters and evaluated the efficiency of our signature scheme through real time implementation. First, we estimate the security parameters by considering attacks mentioned in Section 4, namely direct attack and minrank attack. As for 128-bit security level parameters, we tried out different values of s and confirmed the complexity of both attacks exceed 128 bits. When we choose $s = 5, q = 256$, direct attack requires around 2^{159} computation times and minrank attack requires 2^{157} computation times. Similarly, we select security parameters for 192-bit and 256-bit security level. We confirmed $s = 7, q = 256$ and $s = 8, q = 256$ achieve 192-bit and 256-bit security independently.

Table 2 Parameters of different security level

Security	(q,s,n,m)	Pk size(kB)	Sig. size(kB)
2^{128}	$(256,5,50,25)$	31.12	0.04
2^{196}	$(256,7,98,49)$	232.12	0.09
2^{256}	$(256,8,128,64)$	516.00	0.12

Security	(q,s,n,m)	Sig. time(s)	Ver. time(s)
2^{128}	$(256,5,50,25)$	0.010	0.010
2^{196}	$(256,7,98,49)$	0.030	0.040
2^{256}	$(256,8,128,64)$	0.060	0.100

5.2 Key Sizes

By using those parameters, we can calculate the public key size and signature size immediately. For instance, we use $s = 5, q = 256$ as our 128-bit secure parameter, the public key size in this case can be computed as

$$m \times (\frac{n(n-1)}{2} + n + 1) \times \log_2 q =$$
$$25 \times 1276 \times \log_2 256 = 255,200 \text{ bits}$$

which is around 31kB of storage. We can do similar operations to get public key size of 196-bit and 256-bit security level which are listing in table 2 in the following.

We made experiments using Intel(R) Xeon(R) Gold 6130 CPU 2.10GHz system and MAGMA V2.24-8. From table 2, our signature scheme reveals advantages that the public key size and signature length are small compared to UOV which public key size is above 1.1mB[12]. Also, based on the special structure in our new scheme, the computation speed is faster than UOV and ABC.

6. Conclusion and Future Work

In this paper, we proposed a new family of signature scheme inspired from UOV signature scheme using similar structure of Simple Matrix encryption scheme. We gave out the idea of designing trapdoor and analyzed the security against some attacks. As we see in section 4 and 5, the advantage of this new signature scheme is that the public key size is small and signature generation speed is fast. Also, this new signature scheme can be modified into different applications by changing the central map. At the end of the article, we put forward some practical parameters for implementation.

However, we know that there are many splendid signature schemes in MPKC such as HFEv-[14] which has very short public key size. To compress the public key size, many works are needed to do. A current idea is applying the compression method in LUOV[1] to our new scheme.

References

[1] Ward Beullens, Bart Preneel, Alan Szepieniec, and Frederik Vercauteren. LUOV, signature scheme proposal for NIST PQC project. NIST PQC Submission, imec-COSIC KU Leuven, 2017.

[2] Lily Chen, Stephen Jordan, Yi-Kai Liu, Dustin Moody, Rene Peralta, Ray Perlner, and Daniel Smith-Tone. Report on post-quantum cryptography. NIST Interagency Report 8105, National Institute of Standards and Technology, 2016.

[3] Nicolas Courtois, Alexander Klimov, Jacques Patarin, and Adi Shamir. Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In *Advances in Cryptology – EUROCRYPT 2000*, volume 1807 of *LNCS*, pages 392–407. Springer, 2000.

[4] Adama Diene and Yahya Yusuf. A multivariate signature based on block matrix multiplication. 04 2020.

[5] Jintai Ding, Lei Hu, Xuyun Nie, Jianyu Li, and John Wagner. High order linearization equation (hole) attack on multivariate public key cryptosystems. In Tatsuaki Okamoto and Xiaoyun Wang, editors, *Public Key Cryptography – PKC 2007*, pages 233–248, Berlin, Heidelberg, 2007. Springer Berlin Heidelberg.

[6] Jintai Ding and Dieter Schmidt. Rainbow, a new multivariate polynomial signature scheme. In *Applied Cryptography and Network Security – ACNS 2005*, volume 3531 of *LNCS*, pages 164–175. Springer, 2005.

[7] Jintai Ding, Bo-Yin Yang, CHia-Hsin Owen Chen, Ming-Shing Chen, and Chen-Mou Cheng. New differential-algebraic attacks and reparametrization of rainbow. In *Applied Cryptography and Network Security – ACNS 2008*, volume 5037 of *LNCS*, pages 242–257. Springer, 2008.

[8] Jean-Charles Faugère. A new efficient algorithm for computing Gröbner bases (F4). *Journal of Pure and Applied Algebra*, 139(1):61 – 88, 1999.

[9] Jean Charles Faugère. A new efficient algorithm for computing Gröbner Bases without reduction to zero (F5). In *ISSAC 2002*, pages 75–83. ACM, 2002.

[10] Jean-Charles Faugère, Françoise Levy-dit Vehel, and Ludovic" Perret. Cryptanalysis of minrank. In David Wagner, editor, *Advances in Cryptology – CRYPTO 2008*, pages 280–296, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[11] Yasuhiko Ikematsu, Ray Perlner, Daniel Smith-Tone, Tsuyoshi Takagi, and Jeremy Vates. HFERP - a new multivariate encryption scheme. In *Post-Quantum Cryptography – PQCrypto 2018*, volume 10786 of *LNCS*, pages 396–416. Springer, 2018.

[12] Aviad Kipnis, Jacques Patarin, and Louis Goubin. Unbalanced oil and vinegar signature schemes. In *Advances in Cryptology – EUROCRYPT '99*, volume 1592 of *LNCS*, pages 206–222. Springer, 1999.

[13] Tsutomu Matsumoto and Hideki Imai. Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In *Advances in Cryptology – EUROCRYPT '88*, volume 330 of *LNCS*, pages 419–453. Springer, 1988.

[14] Jacques Patarin. Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms. In *Advances in Cryptology – EUROCRYPT '96*, volume 1070 of *LNCS*, pages 33–48. Springer, 1996.

[15] Jacques Patarin, Nicolas Courtois, and Louis Goubin. FLASH, a fast multivariate signature algorithm. In *Topics in Cryptology – CT-RSA 2001*, volume 2020 of *LNCS*, pages 298–307. Springer, 2001.

[16] P. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[17] Alan Szepieniec, Jintai Ding, and Bart Preneel. Extension field cancellation: A new central trapdoor for multivariate quadratic systems. In *Post-Quantum Cryptography 2016*, volume 9606 of *LNCS*, pages 182–196. Springer, 2016.

[18] Chengdong Tao, Adama Diene, Shaohua Tang, and Jintai Ding. Simple matrix scheme for encryption. In *Post-Quantum Cryptography 2013*, volume 7932 of *LNCS*, pages 231–242. Springer, 2013.

[19] Takanori Yasuda, Xavier Dahan, Yun-Ju Huang, Tsuyoshi Takagi, and Kouichi Sakurai. MQ challenge: Hardness evaluation of solving multivariate quadratic problems. In *Cryptology ePrint Archive: Report 2015/275*, 2015.

[20] Takanori Yasuda and Kouichi Sakurai. A multivariate encryption scheme with Rainbow. In *Information and Communications Security–ICICS 2015*, LNCS, pages 236–251. Springer, 2016.

Appendix

A.1 Toy Example for Simple Matrix Signature Scheme

In this section, we provide a toy example for our new signature scheme. In this case, we let $s = 2, q = 7$ and naturally $n = 8, m = 4$. The matrices A, B, C are randomly selected as:

$$\begin{aligned}
A = &\begin{pmatrix}1&2\\6&6\end{pmatrix} + \begin{pmatrix}6&4\\0&3\end{pmatrix} x_1 + \begin{pmatrix}6&2\\5&1\end{pmatrix} x_2 + \\
&\begin{pmatrix}6&3\\5&3\end{pmatrix} x_3 + \begin{pmatrix}5&0\\2&6\end{pmatrix} x_4 + \\
&\begin{pmatrix}3&1\\1&0\end{pmatrix} x_5 + \begin{pmatrix}0&3\\6&2\end{pmatrix} x_6 + \\
&\begin{pmatrix}0&3\\1&1\end{pmatrix} x_7 + \begin{pmatrix}0&6\\3&5\end{pmatrix} x_8 \\
B = &\begin{pmatrix}2&6\\2&1\end{pmatrix} + \begin{pmatrix}6&2\\1&3\end{pmatrix} x_1 + \begin{pmatrix}6&3\\6&2\end{pmatrix} x_2 + \\
&\begin{pmatrix}5&0\\4&6\end{pmatrix} x_3 + \begin{pmatrix}1&3\\5&3\end{pmatrix} x_4 + \\
&\begin{pmatrix}0&5\\5&1\end{pmatrix} x_5 + \begin{pmatrix}5&2\\5&0\end{pmatrix} x_6 + \\
&\begin{pmatrix}2&6\\3&5\end{pmatrix} x_7 + \begin{pmatrix}2&6\\6&6\end{pmatrix} x_8 \\
C = &\begin{pmatrix}2&4\\4&3\end{pmatrix} + \begin{pmatrix}4&4\\1&6\end{pmatrix} x_1 + \begin{pmatrix}4&4\\5&4\end{pmatrix} x_2 + \\
&\begin{pmatrix}4&5\\0&3\end{pmatrix} x_3 + \begin{pmatrix}4&2\\3&6\end{pmatrix} x_4 + \\
&\begin{pmatrix}3&5\\2&3\end{pmatrix} x_5 + \begin{pmatrix}4&3\\5&3\end{pmatrix} x_6 + \\
&\begin{pmatrix}2&2\\4&5\end{pmatrix} x_7 + \begin{pmatrix}6&1\\0&5\end{pmatrix} x_8.
\end{aligned}$$

The affine transformations T and S are selected as

$$S(x) = \begin{pmatrix}2&4&6&6&4&3&4&5\\2&4&3&0&1&3&1&3\\2&1&1&4&0&5&2&2\\0&4&4&3&6&2&0&1\\4&0&6&0&3&4&6&3\\0&5&5&4&4&1&6&3\\1&1&2&1&0&3&1&3\\2&0&5&6&0&5&2&2\end{pmatrix} x$$

$$T(y) = \begin{pmatrix}2&5&1&1\\0&5&6&4\\2&4&1&4\\2&5&6&6\end{pmatrix} y.$$

By choosing the secret key, we compute the public key using

the structure in Section 3 and the result is:

$$p_1 = X^T \begin{pmatrix} 5 & 1 & 3 & 4 & 2 & 2 & 6 & 1 & 1 \\ 1 & 6 & 3 & 0 & 6 & 5 & 6 & 6 & 2 \\ 3 & 3 & 0 & 2 & 3 & 2 & 0 & 5 & 3 \\ 4 & 0 & 2 & 6 & 3 & 5 & 1 & 2 & 6 \\ 2 & 6 & 3 & 3 & 4 & 5 & 4 & 5 & 5 \\ 2 & 5 & 2 & 5 & 5 & 5 & 5 & 3 & 2 \\ 6 & 6 & 0 & 1 & 4 & 5 & 0 & 0 & 2 \\ 1 & 6 & 5 & 2 & 5 & 3 & 0 & 4 & 5 \\ 1 & 2 & 3 & 6 & 5 & 2 & 2 & 5 & 1 \end{pmatrix} X$$

$$p_2 = X^T \begin{pmatrix} 0 & 0 & 2 & 2 & 3 & 3 & 3 & 1 & 3 \\ 0 & 2 & 4 & 5 & 2 & 4 & 3 & 1 & 2 \\ 2 & 4 & 4 & 6 & 4 & 0 & 5 & 5 & 2 \\ 2 & 5 & 6 & 0 & 3 & 2 & 5 & 6 & 1 \\ 3 & 2 & 4 & 3 & 0 & 5 & 5 & 0 & 6 \\ 3 & 4 & 0 & 2 & 5 & 5 & 4 & 0 & 0 \\ 3 & 3 & 5 & 5 & 5 & 4 & 3 & 3 & 6 \\ 1 & 1 & 5 & 6 & 0 & 0 & 3 & 3 & 3 \\ 3 & 2 & 2 & 1 & 6 & 0 & 6 & 3 & 0 \end{pmatrix} X$$

$$p_3 = X^T \begin{pmatrix} 5 & 6 & 4 & 2 & 6 & 1 & 3 & 4 & 2 \\ 6 & 1 & 4 & 4 & 0 & 0 & 1 & 6 & 1 \\ 4 & 4 & 2 & 1 & 2 & 6 & 4 & 5 & 2 \\ 2 & 4 & 1 & 6 & 2 & 4 & 5 & 0 & 1 \\ 6 & 0 & 2 & 2 & 4 & 3 & 3 & 4 & 0 \\ 1 & 0 & 6 & 4 & 3 & 1 & 0 & 6 & 2 \\ 3 & 1 & 4 & 5 & 3 & 0 & 5 & 3 & 4 \\ 4 & 6 & 5 & 0 & 4 & 6 & 3 & 1 & 0 \\ 2 & 1 & 2 & 1 & 0 & 2 & 4 & 0 & 0 \end{pmatrix} X$$

$$p_4 = X^T \begin{pmatrix} 3 & 2 & 6 & 2 & 2 & 3 & 6 & 3 & 3 \\ 2 & 3 & 2 & 3 & 6 & 3 & 4 & 6 & 2 \\ 6 & 2 & 4 & 1 & 0 & 1 & 3 & 5 & 1 \\ 2 & 3 & 1 & 5 & 1 & 2 & 0 & 5 & 5 \\ 2 & 6 & 0 & 1 & 4 & 1 & 5 & 0 & 4 \\ 3 & 3 & 1 & 2 & 1 & 3 & 2 & 5 & 0 \\ 6 & 4 & 3 & 0 & 5 & 2 & 1 & 0 & 2 \\ 3 & 6 & 5 & 5 & 0 & 5 & 0 & 2 & 0 \\ 3 & 2 & 1 & 5 & 4 & 0 & 2 & 0 & 6 \end{pmatrix} X$$

where $X = (x_1, \ldots, x_8, 1)^T$ and symbol T represents transpose operation. To generate signatures, we randomly choose a message:

$$\mathbf{m} = (2, 2, 5, 4)^T.$$

In first step, we compute the formula $\mathbf{y} = T^{-1}(\mathbf{m})$:

$$\mathbf{y} = (3, 5, 1, 5)^T.$$

To find out a solution of $F(\mathbf{y}) = \mathbf{x}$, we randomly pick a constant matrix D:

$$D = \begin{pmatrix} 5 & 3 \\ 2 & 1 \end{pmatrix}$$

and then solve the linear system:

$$B(x_1, \ldots, x_n) = D$$
$$A(x_1, \ldots, x_n)D + DC(x_1, \ldots, x_n) = Y$$

Easily, we obtain the solution

$$\mathbf{x} = (6, 3, 6, 3, 2, 6, 2, 4)^T.$$

In final step, we compute $\mathbf{s} = S^{-1}(\mathbf{x})$ as a signature:

$$\mathbf{s} = (4, 1, 2, 6, 5, 4, 2, 5)^T$$

To verify the correctness of this signature, obviously we just need to check the equation

$$P(\mathbf{s}) = (p_1, p_2, p_3, p_4)(\mathbf{s}) = \mathbf{m}$$

whether holds.

A.2 Experiment Codes Using Magma

```
//======Parameters======
s:=8;
n:=2*s^2;
m:=s^2;
q:=256;
F<a>:=GF(q);
P<[x]>:=PolynomialRing(F,n,"grevlex");

//======Private Key Generation======
L1:=Random(GL(n,F));
L2:=Random(GL(m,F)); //affine transformations
A:=[Random(GL(s,F)):i in [1..n+1]];
B:=[Random(GL(s,F)):i in [1..n+1]];
C:=[Random(GL(s,F)):i in [1..n+1]];
//secret key generation

X:=Transpose(Matrix(Vector(x)));
Y:=(RMatrixSpace(P,n,n)!L1)*X;
AY:=(&+[(RMatrixSpace(P,s,s)!A[i])*Y[i][1]:i in
    [1..n]])+(RMatrixSpace(P,s,s)!A[n+1]);
BY:=(&+[(RMatrixSpace(P,s,s)!B[i])*Y[i][1]:i in
    [1..n]])+(RMatrixSpace(P,s,s)!B[n+1]);
CY:=(&+[(RMatrixSpace(P,s,s)!C[i])*Y[i][1]:i in
    [1..n]])+(RMatrixSpace(P,s,s)!C[n+1]);
Z:=Transpose(Matrix(Vector(Eltseq(AY*BY+BY*CY)))
    );
pb_key:=RMatrixSpace(P,m,m)!L2*Z; //public key

Ax:=(&+[(RMatrixSpace(P,s,s)!A[i])*x[i]:i in
    [1..n]])+(RMatrixSpace(P,s,s)!A[n+1]);
Bx:=(&+[(RMatrixSpace(P,s,s)!B[i])*x[i]:i in
    [1..n]])+(RMatrixSpace(P,s,s)!B[n+1]);
Cx:=(&+[(RMatrixSpace(P,s,s)!C[i])*x[i]:i in
    [1..n]])+(RMatrixSpace(P,s,s)!C[n+1]);
Bc:=Matrix(m,n,[[MonomialCoefficient(Eltseq(Bx)[
    i],x[j]):j in [1..n]]:i in [1..m]]);

//======Signing and Verifying======
t1:=Cputime(); //Starting time
M:=Random(VectorSpace(F,m)); //message
M1:=L2^-1*Transpose(Matrix(M));
repeat
        D:=Matrix(s,s,Random(VectorSpace(F,m)));
        LHS:=[A[i]*D+D*C[i]:i in [1..n+1]];
        LHSx:=&+[RMatrixSpace(P,s,s)!LHS[i]*x[i
            ]:i in [1..n]];
        LHSx_c:=Matrix(m,n,[[MonomialCoefficient
            (Eltseq(LHSx)[i],x[j]):j in [1..n]]:
            i in [1..m]]);
        linear_c:=VerticalJoin(Bc,LHSx_c);
until Rank(linear_c) eq n;
const:=Eltseq(D-B[n+1]) cat [M1[i][1]-Eltseq(LHS
    [n+1])[i]:i in [1..m]];;
S1:=linear_c^-1*Transpose(Matrix(Vector(const)))
    ;
S:=Eltseq(L1^-1*S1); t2:=Cputime();
Evaluate(pb_key,S); t3:=Cputime();
Sign_time:=t2-t1; //Signature generation time
Ver_time:=t3-t2; //Verification time
```