Finite Fields and Their Applications 35 (2015) 352–368

Contents lists available at ScienceDirect

Finite Fields and Their Applications

www.elsevier.com/locate/ffa

Simple Matrix – A Multivariate Public Key Cryptosystem (MPKC) for Encryption

Chengdong Tao[a], Hong Xiang[b], Albrecht Petzoldt[c], Jintai Ding[b,d,*]

[a] *South China University of Technology, China*
[b] *ChongQing University, China*
[c] *Technische Universität Darmstadt, Germany*
[d] *University of Cincinnati, OH, USA*

ARTICLE INFO

Article history:
Received 27 February 2015
Received in revised form 3 June 2015
Accepted 4 June 2015
Available online 29 June 2015
Communicated by Neal Koblitz

MSC:
20.040

Keywords:
Multivariate cryptography
Public key cryptography
Encryption schemes
Rank attacks

ABSTRACT

Multivariate cryptography is one of the main candidates to guarantee the security of communication in the presence of quantum computers. While there exist a large number of secure and efficient multivariate signature schemes, the number of practical multivariate encryption schemes is somewhat limited. In this paper we present our results on creating a new multivariate encryption scheme, which is an extension of the original SimpleMatrix encryption scheme of PQCrypto 2013. Our scheme allows fast en- and decryption and resists all known attacks against multivariate cryptosystems. Furthermore, we present a new idea to solve the decryption failure problem of the original SimpleMatrix encryption scheme.

1. Introduction

Cryptographic techniques are an essential tool to guarantee the security of communication in the modern society. Today, the security of nearly all of the cryptographic schemes

[*] Corresponding author at: University of Cincinnati, OH, USA.
E-mail address: Jintai.Ding@gmail.com (J. Ding).

http://dx.doi.org/10.1016/j.ffa.2015.06.001
1071-5797/© 2015 Elsevier Inc. All rights reserved.

used in practice is based on number theoretic problems such as factoring large integers and solving discrete logarithms. The best known schemes in this area are RSA [19], DSA [13] and ECC. However, schemes like these will become insecure as soon as large enough quantum computers arrive. The reason for this is Shor's algorithm [20], which solves number theoretic problems such as integer factorization and discrete logarithms in polynomial time on a quantum computer. Therefore, one needs alternatives to those classical public key schemes which are based on mathematical problems not affected by quantum computer attacks.

Besides lattice, code and hash based cryptosystems, multivariate cryptography is one of the main candidates for this [1]. Multivariate schemes are very fast and require only modest computational resources, which makes them attractive for the use on low cost devices such as smart cards and RFID chips [2,3]. However, while there exist many practical multivariate signature schemes [7,12,16], the number of efficient and secure multivariate encryption schemes is somewhat limited.

In this paper we present our results on creating a new multivariate encryption scheme called the SimpleMatrix encryption scheme. The scheme allows fast en- and decryption and resists all known attacks against multivariate schemes.

After describing the basic SimpleMatrix encryption scheme of PQCrypto 2013 [21], we present a new improved version of the scheme. Compared to the original SimpleMatrix scheme, our scheme reduces the probability of decryption failures and speeds up the decryption process further.

The rest of the paper is organized as follows. In Section 2 we give an introduction into the area of multivariate cryptography. Section 3 describes the MinRank attack and its application to the HFE cryptosystem which is one of the most famous multivariate encryption schemes. Section 4 then introduces the SimpleMatrix scheme in its basic form, whereas Section 5 presents our improvements of the original scheme. In Section 6 we give a detailed security analysis of the improved SimpleMatrix scheme. Section 7 describes the parameter selection and Section 8 deals with the efficiency of our scheme and gives a practical implementation. Finally, Section 9 concludes the paper.

2. Multivariate cryptography

Let $\mathbb{F} = \mathbb{F}_q$ be a finite field with q elements. The basic objects of multivariate cryptography are systems of multivariate quadratic polynomials (see equation (1)).

$$\begin{aligned}
p^{(1)}(x_1,\ldots,x_n) &= \sum_{i=1}^{n}\sum_{j=i}^{n} p_{ij}^{(1)} \cdot x_i x_j + \sum_{i=1}^{n} p_i^{(1)} \cdot x_i + p_0^{(1)} \\
p^{(2)}(x_1,\ldots,x_n) &= \sum_{i=1}^{n}\sum_{j=i}^{n} p_{ij}^{(2)} \cdot x_i x_j + \sum_{i=1}^{n} p_i^{(2)} \cdot x_i + p_0^{(2)} \\
&\ \vdots \\
p^{(m)}(x_1,\ldots,x_n) &= \sum_{i=1}^{n}\sum_{j=i}^{n} p_{ij}^{(m)} \cdot x_i x_j + \sum_{i=1}^{n} p_i^{(m)} \cdot x_i + p_0^{(m)} \qquad (1)
\end{aligned}$$

Encryption

$$\begin{tikzcd}
\mathbf{d} \in \mathbb{F}^n \arrow[r, "\mathcal{P}"] & \mathbf{c} \in \mathbb{F}^m \arrow[d, "\mathcal{S}^{-1}"] \\
\mathbf{x} \in \mathbb{F}^n \arrow[u, "\mathcal{T}^{-1}"] & \mathbf{y} \in \mathbb{F}^m \arrow[l, "\mathcal{F}^{-1}"']
\end{tikzcd}$$

Decryption

Fig. 1. General workflow of multivariate encryption schemes.

In the polynomials $p^{(1)}, \ldots, p^{(m)}$, all the coefficients $p_{ij}^{(k)}$, $p_i^{(k)}$ and variables are elements of the field $\mathbb{F}$. The security of multivariate schemes is based on the

Problem MQ. Given m multivariate quadratic polynomials $p^{(1)}(\mathbf{x}), \ldots, p^{(m)}(\mathbf{x})$ as shown in equation (1), find a vector $\bar{\mathbf{x}} = (\bar{x}_1, \ldots, \bar{x}_n)$ such that $p^{(1)}(\bar{\mathbf{x}}) = \ldots = p^{(m)}(\bar{\mathbf{x}}) = 0$.

The MQ problem (for $m \approx n$) is proven to be NP-hard even for quadratic polynomials over the field GF(2) [11].

To build a public key cryptosystem on the basis of the MQ problem, one starts with an easily invertible quadratic map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$ (central map). To hide the structure of $\mathcal{F}$ in the public key, one composes it with two invertible affine (or linear) maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$. The *public key* is therefore given by $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$. The *private key* consists of $\mathcal{S}$, $\mathcal{F}$ and $\mathcal{T}$ and therefore allows to invert the public key.

In this paper we concentrate on multivariate encryption schemes. The standard encryption/decryption process works as shown in Fig. 1.

Encryption: To encrypt a message $\mathbf{d} \in \mathbb{F}^n$, one simply computes $\mathbf{c} = \mathcal{P}(\mathbf{d})$. The ciphertext of the message $\mathbf{d}$ is $\mathbf{c} \in \mathbb{F}^m$.

Decryption: To decrypt the ciphertext $\mathbf{c} \in \mathbb{F}^m$, one computes recursively $\mathbf{y} = \mathcal{S}^{-1}(\mathbf{c})$, $\mathbf{x} = \mathcal{F}^{-1}(\mathbf{y})$ and $\mathbf{d} = \mathcal{T}^{-1}(\mathbf{x})$. $\mathbf{d} \in \mathbb{F}^n$ is the plaintext corresponding to the ciphertext $\mathbf{c}$.

Since, for multivariate encryption schemes, we have $m \geq n$, the pre-image of the vector $\mathbf{y}$ under the central map $\mathcal{F}$ and therefore the decrypted plaintext is unique.

There are numerous proposals to create the central map $\mathcal{F}$ of multivariate cryptosystems. These attempts can be divided into two main groups.

For *BigField schemes*, the central map $\bar{\mathcal{F}} : \mathbb{E} \to \mathbb{E}$ is defined over a large extension field $\mathbb{E}$ of $\mathbb{F}$ of degree n. One uses an isomorphism ϕ between $\mathbb{E}$ and the vector space $\mathbb{F}^n$ to transform it into a map $\tilde{\mathcal{F}} = \phi \circ \bar{\mathcal{F}} \circ \phi^{-1} : \mathbb{F}^n \to \mathbb{F}^n$. The public key has the form

$$\mathcal{P} = \mathcal{S} \circ \tilde{\mathcal{F}} \circ \mathcal{T} = \mathcal{S} \circ \phi \circ \bar{\mathcal{F}} \circ \phi^{-1} \circ \mathcal{T}$$

with two invertible linear (or affine) maps $\mathcal{S}$ and $\mathcal{T}$ (see Fig. 2).

$$\begin{tikzcd}
 & \mathbf{X}\in\mathbb{E} \arrow[r, "\bar{\mathcal{F}}"] & \mathbf{Y}\in\mathbb{E} \arrow[d, "\phi"] & \\
\mathbf{d}\in\mathbb{F}^n \arrow[r, "\mathcal{T}"] \arrow[rrr, bend right, "\mathcal{P}"'] & \mathbf{x}\in\mathbb{F}^n \arrow[u, "\phi^{-1}"] \arrow[r, "\tilde{\mathcal{F}}"] & \mathbf{y}\in\mathbb{F}^n \arrow[r, "\mathcal{S}"] & \mathbf{c}\in\mathbb{F}^n
\end{tikzcd}$$

Fig. 2. Construction of BigField schemes.

Algorithm 1 MinRank attack.

```
Input: matrices $P^{(1)}, \dots, P^{(m)}$
Output: Linear combination $\tilde{P} = \sum_{i=1}^{m} \alpha_i \cdot P^{(i)}$ of rank $\le r$
repeat
    Choose randomly a vector $\lambda \in \mathbb{F}^m$ and compute $P = \sum_{i=1}^{m} \lambda_i \cdot P^{(i)}$.
    if Rank $(P) > 1$ and Rank$(P) < n$ then
        Choose randomly a vector $\alpha$ from Ker$(P)$.
        $\tilde{P} \leftarrow \sum_{i=1}^{m} \alpha_i \cdot P^{(i)}$
    end if
until Rank $(\tilde{P}) \le r$
return $\tilde{P}$
```

For *SingleField schemes*, all the computations are performed over a relatively small field $\mathbb{F}$. One chooses an easily invertible quadratic map $\mathcal{F} : \mathbb{F}^n \to \mathbb{F}^m$ and combines $\mathcal{F}$ in the public key with two invertible linear (or affine) maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$, i.e. $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$.

So far, all known constructions for multivariate encryption schemes are BigField constructions. The best known examples for this are MI [15] and HFE [18]. On the other hand, SingleField constructions were only used for multivariate signature schemes such as UOV and Rainbow [7].

In this sense, the SimpleMatrix scheme as proposed in Section 4 is the first multivariate encryption scheme based on a SingleField construction. This enables us to perform all the operations in a relatively small field which makes our scheme much more efficient than for example HFE (see also Section 8). More information on existing multivariate schemes can be found in [5].

3. The MinRank attack

The MinRank attack [10] is a general attack against multivariate cryptosystems. The MinRank problem can be defined as follows.

Problem MinRank. Given m $n \times n$ matrices $P^{(1)}, \dots, P^{(m)}$, find a linear combination $\widetilde{P} = \sum_{i=1}^{m} \alpha_i P^{(i)}$ of minimal rank r.

This problem can be solved as shown in Algorithm 1, whose complexity can be estimated by [10]

$$\text{Complexity}_{\text{MinRank}} = \mathcal{O}(q^{\lceil \frac{m}{n} \rceil \cdot r} \cdot m^3). \tag{2}$$

Here, q is the cardinality of the underlying field $\mathbb{F}$.

3.1. The MinRank attack against HFE

3.1.1. The HFE cryptosystem

The HFE cryptosystem as proposed by Patarin in [18] is one of the best known multivariate encryption schemes. Let p be a prime number, $\mathbb{F} = \mathrm{GF(p)}$ and $\mathbb{E}$ be a degree n extension field of $\mathbb{F}$. Let $\phi : \mathbb{E} \to \mathbb{F}^n$ be the canonical isomorphism between the field $\mathbb{E}$ and the vector space $\mathbb{F}^n$. The central map $\bar{\mathcal{F}}$ of the HFE scheme is a map from $\mathbb{E}$ to itself of the form

$$\bar{\mathcal{F}}(X) = \sum_{i=0}^{r-1} \sum_{j=0}^{r-1} a_{ij} X^{q^i + q^j}, \tag{3}$$

where the coefficients a_{ij} are chosen from the field $\mathbb{E}$ and r is a small constant (to enable efficient decryption).[1] Due to the special structure of the map $\bar{\mathcal{F}}$, the map

$$\tilde{\mathcal{F}} : \mathbb{F}^n \to \mathbb{F}^n, (x_1, \dots, x_n) \mapsto \phi \circ \bar{\mathcal{F}} \circ \phi^{-1}(x_1, \dots, x_n) \tag{4}$$

is a homogeneous quadratic map from $\mathbb{F}^n$ to itself.

To hide the structure of $\tilde{\mathcal{F}}$ in the public key, one combines it with two invertible linear maps $\mathcal{S}$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$. Therefore, the public key of the HFE scheme has the form

$$\mathcal{P}(x_1, \dots, x_n) = \mathcal{S} \circ \tilde{\mathcal{F}} \circ \mathcal{T}(x_1, \dots, x_n) = \mathcal{S} \circ \phi \circ \bar{\mathcal{F}} \circ \phi^{-1} \circ \mathcal{T}(x_1, \dots, x_n). \tag{5}$$

3.1.2. The MinRank attack on HFE

In this paragraph we describe the attack of Kipnis and Shamir [14] against the HFE cryptosystem. The key idea of their attack is to lift the maps $\mathcal{S}$, $\mathcal{T}$ and $\mathcal{P}$ to functions $\mathcal{S}^\star$, $\mathcal{T}^\star$ and $\mathcal{P}^\star$ over the extension field $\mathbb{E}$. Since $\mathcal{S}$ and $\mathcal{T}$ are linear maps, $\mathcal{S}^\star$ and $\mathcal{T}^\star$ have the form

$$\mathcal{S}^\star(X) = \sum_{i=1}^{n-1} s_i \cdot X^{q^i} \quad \text{and} \quad \mathcal{T}^\star(X) = \sum_{i=1}^{n-1} t_i \cdot X^{q^i}, \tag{6}$$

with coefficients s_i and $t_i \in \mathbb{E}$. The function $\mathcal{P}^\star$ can be expressed as

$$\mathcal{P}^\star(X) = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} p^\star_{ij} X^{q^i + q^j} = \underline{X} \cdot P^\star \cdot \underline{X}^T, \tag{7}$$

[1] For the simplicity of our description, we restrict here to a homogeneous map $\bar{\mathcal{F}}$.

where $P^\star = [p^\star_{ij}] \in \mathbb{E}^{n\times n}$ and $\underline{X} = (X^{q^0}, X^{q^1}, \ldots, X^{q^{n-1}})$. Due to the relation $\mathcal{P}^\star(X) = \mathcal{S}^\star \circ \bar{\mathcal{F}} \circ \mathcal{T}^\star(X)$ we get $\mathcal{S}^{\star\ -1} \circ \mathcal{P}^\star(X) = \bar{\mathcal{F}} \circ \mathcal{T}^\star(X)$ and

$$\tilde{P} = \sum_{k=0}^{n-1} s_k \cdot P^{\star k} = W \cdot \bar{F} \cdot W^T \tag{8}$$

with $p^{\star}_{ij}{}^{k} = p^{q^k}_{i-k,j-k}$, $w_{ij} = s^{q^i}_{j-i \bmod n}$ and $\bar{F}$ being the $n \times n$ matrix representing the central map $\bar{\mathcal{F}}$. Note that, due to the special structure of $\bar{\mathcal{F}}$, the only nonzero entries in the matrix $\bar{F}$ are located in the upper left $r \times r$ submatrix.

Since the rank of the matrix $W \cdot \bar{F} \cdot W^T$ is less or equal to r, we can determine the coefficients s_k by solving an instance of the MinRank problem (i.e. by Algorithm 1).

In the context of multivariate public key cryptosystems, the MinRank attack can be used to find linear combinations of the public key polynomials of low rank which then correspond to central polynomials. The MinRank attack is therefore applicable to all multivariate schemes whose central polynomials are of low rank. The key strategy of our construction is therefore to develop a multivariate encryption scheme, whose central polynomials are of relatively high rank.

4. The basic SimpleMatrix encryption scheme

4.1. Description of the scheme

The basic SimpleMatrix encryption scheme as proposed in [21] can be described as follows.

Key generation: Let $\mathbb{F}$ be a finite field with q elements. For a parameter $s \in \mathbb{N}$ we set $n = s^2$ and $m = 2n$ and define three $s \times s$ matrices A, B and C of the form

$$A = \begin{pmatrix} x_1 & \ldots & x_s \\ \vdots & & \vdots \\ x_{(s-1)\cdot s+1} & \ldots & x_n \end{pmatrix},\ B = \begin{pmatrix} b_1 & \ldots & b_s \\ \vdots & & \vdots \\ b_{(s-1)\cdot s+1} & \ldots & b_n \end{pmatrix},\ C = \begin{pmatrix} c_1 & \ldots & c_s \\ \vdots & & \vdots \\ c_{(s-1)\cdot s+1} & \ldots & c_n \end{pmatrix}.$$

Here, $x_1, \ldots, x_n$ are the linear monomials of the multivariate polynomial ring $\mathbb{F}[x_1, \ldots, x_n]$, whereas $b_1, \ldots, b_n$ and $c_1, \ldots, c_n$ are randomly chosen linear combinations of $x_1, \ldots, x_n$.

One computes $E_1 = A \cdot B$ and $E_2 = A \cdot C$. The central map $\mathcal{F}$ of the scheme consists of the m components of E_1 and E_2.

The *public key* of the scheme is the composed map $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T} : \mathbb{F}^n \to \mathbb{F}^m$ with two randomly chosen invertible linear maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$, the *private key* consists of the matrices B and C and the linear maps $\mathcal{S}$ and $\mathcal{T}$.

Encryption: To encrypt a message $\mathbf{d} \in \mathbb{F}^n$, one simply computes $\mathbf{c} = \mathcal{P}(\mathbf{d}) \in \mathbb{F}^m$.

Decryption: To decrypt a ciphertext $\mathbf{c} \in \mathbb{F}^m$, one has to perform the following three steps.

1. Compute $\mathbf{y} = \mathcal{S}^{-1}(\mathbf{c})$. The elements of the vector $\mathbf{y} \in \mathbb{F}^m$ are written into matrices $\bar{E}_1$ and $\bar{E}_2$ as follows.

$$\bar{E}_1 = \begin{pmatrix} y_1 & \cdots & y_s \\ \vdots & & \vdots \\ y_{(s-1)\cdot s+1} & \cdots & y_n \end{pmatrix}, \ \bar{E}_2 = \begin{pmatrix} y_{n+1} & \cdots & y_{n+s} \\ \vdots & & \vdots \\ y_{n+(s-1)\cdot s+1} & \cdots & y_m \end{pmatrix}.$$

2. In the second step one has to find a vector $\mathbf{x} = (x_1, \ldots, x_n)$ such that $\mathcal{F}(\mathbf{x}) = \mathbf{y}$. To do this, one has to distinguish four cases:
   - If $\bar{E}_1$ is invertible, one considers the equation $B \cdot \bar{E}_1^{-1} \cdot \bar{E}_2 - C = 0$. Therefore one gets n linear equations in the n variables $x_1, \ldots, x_n$.
   - If $\bar{E}_1$ is not invertible, but $\bar{E}_2$ is invertible, one considers the equation $C \cdot \bar{E}_2^{-1} \cdot \bar{E}_1 - B = 0$. One gets n linear equations in the n variables.
   - If none of $\bar{E}_1$ and $\bar{E}_2$ is invertible, but $\bar{A} = A(\mathbf{x})$ is invertible, one considers the relations $\bar{A}^{-1} \cdot \bar{E}_1 - B = 0$ and $\bar{A}^{-1} \cdot \bar{E}_2 - C = 0$. One interprets the elements of $\bar{A}^{-1}$ as new variables $w_1, \ldots, w_n$ and therefore gets m linear equations in the m variables $w_1, \ldots, w_n, x_1, \ldots, x_n$.
   - If none of $\bar{E}_1$, $\bar{E}_2$ and $\bar{A}$ is invertible, there occurs a decryption failure.
3. Finally, one computes the plaintext by $\mathbf{d} = \mathcal{T}^{-1}(x_1, \ldots, x_n)$.

It might happen that the linear systems in the second step have multiple solutions $\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(\ell)}$. In this case one has to perform the third step of the decryption process for each of these solutions to get a set of possible plaintexts $\mathbf{d}^{(1)}, \ldots, \mathbf{d}^{(\ell)}$. By encrypting these plaintexts one can test which of them corresponds to the given ciphertext $\mathbf{c}$.

4.2. Probability of decryption failures

If the matrix $\bar{A}$ occurring in the second step of the decryption process is not invertible, there occurs a decryption failure. Since the probability of $\bar{A}$ being singular is given by

$$1 - (1 - \frac{1}{q^s})(1 - \frac{1}{q^{s-1}}) \cdots (1 - \frac{1}{q}) \approx \frac{1}{q},$$

we can estimate the probability of a decryption failure occurring by $\frac{1}{q}$.

5. The improved new scheme

The basic SimpleMatrix encryption scheme as described in the previous section has two main disadvantages:

- The probability of a decryption failure is (at least for moderate field sizes) relatively large.
- The decryption process is not very efficient.

To overcome these disadvantages, we come up with a new strategy. Instead of using square matrices A, B and C as in the basic version of the scheme, we generalize the scheme to non-square matrices.

5.1. Description of the scheme

Key generation: Let $\mathbb{F}$ be a finite field with q elements and $r, s, u, v, m, n \in \mathbb{N}$ be integers satisfying $m = s \cdot (u+v)$, $s \geq r$ and $(n - r(u+v-s)) \cdot (n - r(u+v-s)+1) \leq 2m$.[2] Set

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1r} \\ a_{21} & a_{22} & \dots & a_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ a_{s1} & a_{s2} & \dots & a_{sr} \end{pmatrix}, \quad B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1u} \\ b_{21} & b_{22} & \dots & b_{2u} \\ \vdots & \vdots & \ddots & \vdots \\ b_{r1} & b_{r2} & \dots & b_{ru} \end{pmatrix}, C = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1v} \\ c_{21} & c_{22} & \dots & c_{2v} \\ \vdots & \vdots & \ddots & \vdots \\ c_{r1} & c_{r2} & \dots & c_{rv} \end{pmatrix},$$

where A is an $s \times r$ matrix, B is an $r \times u$ matrix and C is an $r \times v$ matrix. The elements a_{ij} are randomly chosen from the set $\{x_1, \dots, x_n\}$, while the elements b_{ij} and c_{ij} are randomly chosen linear combinations of $x_1, \dots, x_n$. As in Subsection 4.1 we define $E_1 = A \cdot B$ and $E_2 = A \cdot C$. The central map $\mathcal{F}$ of our improved scheme consists of the $m = s \cdot (u+v)$ components of the matrices E_1 and E_2. Note that each of these components is a homogeneous quadratic polynomial in $\mathbb{F}[x_1, \dots, x_n]$ whose associated quadratic form has a rank close or equal to $2r$.

Additionally one chooses two invertible linear maps $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{T} : \mathbb{F}^n \to \mathbb{F}^n$.

The *public key* of the scheme is the composed map $\mathcal{P} = \mathcal{S} \circ \mathcal{F} \circ \mathcal{T}$, the *private key* consists of the linear maps $\mathcal{S}$ and $\mathcal{T}$ as well as the matrices A, B and C.

Encryption: For a message $\mathbf{d} = (d_1, d_2, \dots, d_n) \in \mathbb{F}^n$, the corresponding ciphertext $\mathbf{c} \in \mathbb{F}^m$ is given by $\mathbf{c} = \mathcal{P}(\mathbf{d})$.

Decryption: To decrypt the ciphertext $\mathbf{c} = (c_1, c_2, \dots, c_m) \in \mathbb{F}^m$, one needs to perform the following three steps:

1. Compute $\mathbf{y} = (y_1, y_2, \dots, y_m) = \mathcal{S}^{-1}(\mathbf{c})$ and set

$$\bar{E}_1 = \begin{pmatrix} y_1 & y_2 & \cdots & y_u \\ y_{u+1} & y_{u+2} & \cdots & y_{2u} \\ \vdots & \vdots & \ddots & \vdots \\ y_{(s-1)u+1} & y_{(s-1)u+2} & \cdots & y_{su} \end{pmatrix} \in \mathbb{F}^{s \times u} \text{ and}$$

[2] The reason why we choose $(n - r(u+v-s)) \cdot (n - r(u+v-s)+1) \leq 2m$ lies in the effectiveness of the decryption process (see Subsection 5.2).

$$\bar{E}_2 = \begin{pmatrix} y_{su+1} & y_{su+2} & \cdots & y_{su+v} \\ y_{su+v+1} & y_{su+v+2} & \cdots & y_{su+2v} \\ \vdots & \vdots & \ddots & \vdots \\ y_{su+(s-1)v+1} & y_{su+(s-1)v+2} & \cdots & y_{su+sv} \end{pmatrix} \in \mathbb{F}^{s \times v}.$$

2. In the second step, we have to find a vector $\mathbf{x} \in \mathbb{F}^n$ such that $\mathcal{F}(\mathbf{x}) = \mathbf{y}$. Let $\bar{A} = A(\mathbf{x})$.
   - If the rank of $\bar{A}$ is r, then there exists an $r \times s$ matrix W such that $W \cdot \bar{A} = I$, where I is the $r \times r$ identity matrix. From $\bar{E}_1 = \bar{A} \cdot B$ and $\bar{E}_2 = \bar{A} \cdot C$ we get $W \cdot \bar{E}_1 = W \cdot \bar{A} \cdot B$, $W \cdot \bar{E}_2 = W \cdot \bar{A} \cdot C$ and therefore $W \cdot \bar{E}_1 = B, W \cdot \bar{E}_2 = C$. We interpret the elements of W as new variables and end up with $r(u+v)$ linear equations in $sr + n$ unknowns. Then we eliminate the sr elements of W from these equations. We obtain roughly $r \cdot (u+v-s)$ linear equations in the variables $x_1, x_2, \ldots, x_n$.

     The dimension of the solution space of this system is usually very small. Solving this system by Gaussian elimination enables us to eliminate most of the unknowns, say Z of them. Then we write these Z variables as linear combinations of the remaining unknown variables and substitute these equations into the central polynomials. By doing so, we obtain a new system of m quadratic equations in the remaining $n-Z$ unknowns. When the number $n-Z$ is small enough, we can solve this system efficiently by the Relinearization algorithm of [12] (see Subsection 5.2).
   - In the case of $\mathrm{Rank}(\bar{A}) < r$, decryption remains an open problem.
3. Compute the plaintext $\mathbf{d} \in \mathbb{F}^n$ by $\mathbf{d} = (d_1, d_2, \ldots, d_n) = \mathcal{T}^{-1}(\mathbf{x})$.

5.2. Solving the quadratic systems

During the second step of the decryption process we have to solve a system of quadratic equations. This system consists of m equations in about $n - r \cdot (u+v-s)$ variables. If the condition

$$(n - r \cdot (u+v-s)) \cdot (n - r \cdot (u+v-s) + 1) \leq 2m \tag{9}$$

is fulfilled, we can solve this system easily by the Relinearization technique [12]

1. Interpret each quadratic monomial $x_i x_j$ as a new variable x_{ij}.
2. Solve the resulting linear system by Gaussian elimination.

If the condition (9) is fulfilled, the linear system will have exactly one solution which coincides with the solution of the quadratic system.[3] The Relinearization technique therefore enables us to find the solution of highly overdetermined systems in polynomial time.

[3] As we find, the quadratic systems occurring during the decryption process of our scheme have often less than $n - r \cdot (u+v-s)$ variables. This enables us to use the Relinearization technique even for some parameter sets in which the condition (9) is not fulfilled. We will use this fact when selecting parameters for our scheme over $\mathrm{GF}(2^{16})$ and $\mathrm{GF}(2^8)$ (see Section 7).

5.3. Variant of the scheme

We can make the matrix A more general by choosing its entries as random linear combinations of the monomials $x_1, \ldots, x_n$. In this case, we can omit the linear transformation $\mathcal{T}$. The other parameters are chosen as above. As in Subsection 5.1 we define $E_1 = A \cdot B$ and $E_2 = A \cdot C$. The central map $\mathcal{F}$ of the scheme consists of the m components of E_1 and E_2.

The *public key* of the scheme is therefore the composed map $\mathcal{P} = \mathcal{S} \circ \mathcal{F}$ with an invertible linear transformation $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$, the *private key* consists of the invertible linear transformation $\mathcal{S}$ and the matrices A, B and C.

The encryption and decryption processes work as shown in Subsection 5.1. However, we can omit step 3 of the decryption process.

5.4. Probability of decryption failures

In the case of the improved SimpleMatrix encryption scheme, a decryption failure occurs if and only if the rank of the matrix $\bar{A}$ is less than r. The probability of this can be computed by

$$1 - (1 - \frac{1}{q^s})(1 - \frac{1}{q^{s-1}}) \cdots (1 - \frac{1}{q^{s-r+1}}) \approx \frac{1}{q^{s-r+1}}.$$

Therefore, the probability of a decryption failure occurring can be estimated by $\frac{1}{q^{s-r+1}}$. By choosing the parameters s and r of our scheme in an appropriate way, we can therefore reduce the probability of decryption failures to a negligible value.

6. Security analysis

We analyzed the security of the improved SimpleMatrix encryption scheme against all known attacks against multivariate schemes, including

- (High order) linearization equation attacks
- Rank attacks and
- direct/algebraic attacks.

6.1. (High order) linearization equation attack

The linearization equation attack was first discussed in [17] to attack the $C^\star$ scheme of Matsumoto and Imai [15]. Later, the high order linearization equation attack was proposed to attack the MFE cryptosystem [6]. We use this method to attack our scheme.

The relations $E_1 = A \cdot B$ and $E_2 = A \cdot C$ yield the existence of a polynomial g_1 with $\deg(g_1) \leq r$, such that $B \cdot g_1(\bar{E}_1) \cdot \bar{E}_2 = C \cdot \det(\bar{E}_1)$. Therefore, the plaintext and the ciphertext satisfy the equation:

$$\sum_{i_0=1}^{n} \sum_{1\le i_1 \le \ldots \le i_r \le m} \mu_{i_0,i_1,\ldots,i_r} d_{i_0} c_{i_1} \ldots c_{i_s} +$$

$$+ \sum_{i_0=1}^{n} \sum_{1\le i_1 \le \ldots i_{r-1} \le m} \nu_{i_0,i_1,\ldots,i_{r-1}} d_{i_0} c_{i_1} \ldots c_{i_{r-1}} + \ldots$$

$$+ \sum_{i_0=1}^{n} \gamma_{i_0} d_{i_0} + \sum_{i_1=1}^{m} \xi_{i_1} c_{i_1} + \theta = 0, \tag{10}$$

which means that we can derive linearization equations of order $n+1$. The coefficients $\mu_{i_0,i_1,\ldots,i_r}, \nu_{i_0,i_1,\cdots,i_{r-1}}, \ldots, \gamma_{i_0}, \xi_{i_1}, \theta$ hereby are (so far unknown) elements of the field $\mathbb{F}$. The number of these coefficients is

$$n \sum_{j=0}^{r} \binom{m}{j} + m + 1 = n\binom{m+r}{r} + m + 1.$$

Using the public key we can generate many plaintext/ciphertext pairs. By substituting these plaintext/ciphertext pairs into equation (10), we get $n\binom{m+r}{r}+m+1$ linear equations in $n\binom{m+r}{r}+m+1$ variables. However, the computation complexity of solving this system is $\left(n\binom{m+r}{r}+m+1\right)^{\omega}$, where $\omega = 3$ in the usual Gaussian elimination algorithm and $\omega = 2.3766$ in improved algorithms. This shows that, for reasonably chosen parameters, high order linearization attacks against the improved SimpleMatrix encryption scheme are completely impractical.

6.2. Rank attacks

There are two different methods of using the rank attack. The first one is called the MinRank attack or LowRank attack as proposed by Goubin et al. in [10]. The other one is called the HighRank attack [4]. In this subsection we analyze the complexity of these two attacks against our scheme.

In the case of the SimpleMatrix scheme, the components of the central map $\mathcal{F}$ and the public key $\mathcal{P}$ are homogeneous quadratic polynomials in $\mathbb{F}[x_1, \ldots, x_n]$. Let Q_i and $\bar{Q}_i$ $(i = 1, \ldots, m)$ be the symmetric matrices associated with the i-th component of $\mathcal{F}$ and $\mathcal{P}$ respectively. Note that, for underlying fields of characteristic 2, the diagonal elements of these matrices are 0. In the case of the SimpleMatrix scheme, the rank of the matrices Q_i is obviously bounded by $2r$.

The goal of the MinRank attack is to find a vector $\mathbf{t} = (t_1, t_2, \ldots, t_m) \in \mathbb{F}^m$ such that the rank of the matrix $\tilde{Q} = \sum_{i=1}^{m} t_i \cdot \bar{Q}_i$ is less or equal to $2r$. Such a matrix $\tilde{Q}$ corresponds to a central polynomial. By finding m of these matrices of low rank the attacker can therefore recover the affine map $\mathcal{S}$ and therefore the private key of the scheme.

In order to find such a matrix $\tilde{Q}$ of low rank, the attacker can proceed as shown in Algorithm 1.

The complexity of the MinRank attack against the improved SimpleMatrix scheme can be estimated by

$$\mathcal{O}(q^{\lceil \frac{m}{n} \rceil \cdot 2r} m^3) = \mathcal{O}(q^{4r} \cdot r^6). \tag{11}$$

For the HighRank attack, we form an arbitrary linear combination $\tilde{Q} = \sum_{i=1}^{m} \alpha_i \cdot \bar{Q}_i$ and find $V = \mathrm{Ker}(\tilde{Q})$. If $\tilde{Q}$ has a nontrivial kernel, we set $(\sum_{i=1}^{m} \lambda_i \cdot \bar{Q}_i) \cdot V = 0$ and check if the solution set $\hat{V}$ for the λ_i has dimension $n - 2r$. If this is true, V is, with a certain probability, a subspace of $\mathcal{T}^{-1}(\mathcal{O})$ where $\mathcal{O} = \{\mathbf{x} = (x_1, \ldots, x_n) : x_1 = \ldots = x_{n-2r} = 0\}$.

We can therefore use the HighRank attack to recover the secret linear transformation $\mathcal{T}$ and therefore the private key of our scheme. The complexity of the HighRank attack against the improved SimpleMatrix encryption scheme can be estimated by

$$\mathcal{O}(n^6 \cdot q^{2r}) = \mathcal{O}(r^{12} \cdot q^{2r}). \tag{12}$$

As equations (11) and (12) show, Rank attacks against the improved SimpleMatrix scheme are completely impractical.

6.3. Direct attacks

The most straightforward way to attack a multivariate encryption scheme is by trying to solve the public system $\mathcal{P}(\mathbf{d}) = \mathbf{c}$ for $\mathbf{c}$ (plaintext recovery attack). To do this, an attacker can use an algorithm like XL or a Gröbner basis method such as F_4 or F_5 [9].

To analyze the security of the improved SimpleMatrix encryption scheme against direct attacks, we performed a number of experiments with MAGMA v.18-9, which contains an efficient implementation of Faugères F_4 algorithm. In particular, we considered the public keys of instances of the improved SimpleMatrix scheme over GF(2^8), GF(2^{16}) and GF(2^{32}) whose parameters fulfilled the relations

- $(r, s, u, v, m, n) = (r, r+1, r, r, 2 \cdot r \cdot (r+1), r \cdot (r+1))$ for $\mathbb{F} = \mathrm{GF}(2^{32})$,
- $(r, s, u, v, m, n) = (r, r+2, r+2, r+2, 2 \cdot (r+2)^2, (r+2)^2)$ for $\mathbb{F} = \mathrm{GF}(2^{16})$ and
- $(r, s, u, v, m, n) = (r, r+3, r+4, r+4, 2(r+3)(r+4), r(r+8))$ for $\mathbb{F} = \mathrm{GF}(2^8)$.[4]

The experiments where performed on a server with 128 GB RAM and 16 AMD Opteron CPUs with 2.8 GHz (using a single core for each experiment). The results of the experiments are shown in Table 1.

As the table shows, the public systems of the improved SimpleMatrix encryption scheme can be solved significantly faster than random systems (especially for the case of GF(2^{16}) and GF(2^8)). We therefore made an extrapolation to estimate the complexity

[4] The reason, why we made our experiments for these two parameter sets, can be found in the next section (Section 7).

Table 1
Running time of the direct attack against our improved ABC scheme.

	field	$\mathrm{GF}(2^{32})$			$\mathrm{GF}(2^{16})$		
	parameters (r, s, u, v) (m, n)	$(3, 4, 3, 3)$ $(24, 12)$	$(4, 5, 4, 4)$ $(40, 20)$	$(5, 6, 5, 5)$ $(60, 30)$	$(2, 4, 4, 4)$ $(32, 16)$	$(3, 5, 5, 5)$ $(50, 25)$	$(4, 6, 6, 6)$ $(72, 36)$
our scheme	time (s)	0.63	2133	–	0.23	146.9	87,638
	d_{reg}	4	5	6	3	4	5
	memory (MB)	17.0	621	ooM[a]	17.8	241	3281
for comparison: random system	time (s)	1.4	4151	–	32.9	29,202	–
	d_{reg}	4	5	6	5	6	7
	memory (MB)	18.2	1157	–	76.7	23,181	–

[a] ooM = out of memory.

of solving the public systems of the improved ABC scheme for larger parameters. By doing so, we obtained the following formulas

$$\begin{aligned} \mathrm{Compl}(\mathrm{GF}(2^{32}), r, r+1, r, r, 2r \cdot (r+1), r \cdot (r+1)) &\approx 11.5 \cdot r + 5.5, \\ \mathrm{Compl}(\mathrm{GF}(2^{16}), r, r+2, r+2, r+2, 2 \cdot (r+2)^2, (r+2)^2) &\approx 9 \cdot r + 13.5, \\ \mathrm{Compl}(\mathrm{GF}(2^{8}), r, r+3, r+4, r+4, (r+3)(r+4), r(r+8)) &\approx 9 \cdot r + 13.0. \end{aligned} \tag{13}$$

The parameter proposals made in Section 7 are based on this heuristic.

7. Parameter choice

For the fields $\mathrm{GF}(2^{32})$ and $\mathrm{GF}(2^{16})$ we choose the parameters of the improved SimpleMatrix encryption scheme in such a way that the probability of a decryption failure occurring is less than 2^{-40}. We therefore need

- $s - r = 1$ for $\mathbb{F} = \mathrm{GF}(2^{32})$ and
- $s - r = 2$ for $\mathbb{F} = \mathrm{GF}(2^{16})$.

Furthermore we want for security reasons that $m \leq 2n$. With this equation and $u = v$, the condition

$$(n - r \cdot (2 \cdot u - s)) \cdot (n - r \cdot (2 \cdot u - s) + 1) \leq 2 \cdot m \tag{14}$$

needed for the efficiency of the decryption process yields

- $u \geq r$ for $\mathbb{F} = \mathrm{GF}(2^{32})$ and
- $u \geq r + 3$ for $\mathbb{F} = \mathrm{GF}(2^{16})$.

As we found by experiments, the decryption remains still possible when using $u = r + 2$ over $\mathrm{GF}(2^{16})$ (see footnote 3).

To get efficient parameters for the improved SimpleMatrix scheme over $\mathrm{GF}(2^8)$, we have to slighten the above conditions a bit.

Table 2
Proposed parameters and resulting key sizes for the improved SimpleMatrix encryption scheme.

proposed security level (bit)	parameters $\mathbb{F}$, (r, s, u, v, m, n)	input size (bit)	output size (bit)	public key size (kB)	private key size (kB)	probability of decryption error
80	GF(2^{32}), $(7, 8, 7, 7, 112, 56)$	1792	3584	698	82.7	2^{-64}
	GF(2^{16}), $(8, 10, 10, 10, 200, 100)$	1600	3200	1934	129.0	2^{-48}
	GF(2^{8}), $(8, 11, 12, 12, 264, 128)$	1008	2112	2062	84.0	2^{-32}
90	GF(2^{32}), $(8, 9, 8, 8, 144, 72)$	2304	4608	1478	137.3	2^{-64}
	GF(2^{16}), $(9, 11, 11, 11, 242, 121)$	1936	3872	3489	189.9	2^{-48}
	GF(2^{8}), $(9, 12, 13, 13, 312, 153)$	1200	2496	3451	117.0	2^{-32}
100	GF(2^{32}), $(9, 10, 9, 9, 180, 90)$	2880	5760	2879	215.2	2^{-64}
	GF(2^{16}), $(10, 12, 12, 12, 288, 144)$	2304	4608	5873	270.1	2^{-48}
	GF(2^{8}), $(10, 13, 14, 14, 364, 180)$	1408	2912	5537	160.0	2^{-32}

In particular, we set

- pr(decryption failure) $\leq 2^{-32}$
- $m \leq 2.08 \cdot n$ and
- $u = v$.

We therefore obtain the three parameter sets

- $(r, s, u, v, m, n) = (r, r + 1, r, r, 2r(r + 1), r(r + 1))$ for $\mathbb{F} = \mathrm{GF}(2^{32})$,
- $(r, s, u, v, m, n) = (r, r + 2, r + 2, r + 2, 2(r + 2)^2, (r + 2)^2)$ for $\mathbb{F} = \mathrm{GF}(2^{16})$ and
- $(r, s, u, v, m, n) = (r, r + 3, r + 4, r + 4, 2(r + 3)(r + 4), r(r + 8))$ for $\mathbb{F} = \mathrm{GF}(2^{8})$.

Note that these are exactly the parameter sets used for our experiments in Subsection 6.3.

Table 2 shows, for different levels of security and field sizes, the resulting key and ciphertext sizes of our scheme. As can be seen from the table, the field GF(2^{32}) is most suitable for our scheme. When using smaller fields, the size of the public key is very large.

7.1. Reducing the private key size

In order to decrease the size of the private key, one may think of choosing the matrices B and C in such a way that their entries are randomly selected sparse linear functions or even monomials. However, this might lead to a sparse central map $\mathcal{F}$. One therefore has to ensure that the central polynomials are not so sparse that they have hidden UOV structures. In this case an attacker might use the UOV Reconciliation attack [8] to find these structures. It is therefore not a good idea to choose the elements of the matrices B and C to be monomials, since such a distinguished feature is in general not desired.

In this subsection we consider the question how sparse the linear combinations in the matrices B and C can be chosen without weakening the security of our scheme. We look at this question in the context of our variant of the improved SimpleMatrix scheme (see Subsection 5.3). Therefore, the elements of the matrix A are randomly chosen linear combinations of the monomials $x_1, \ldots, x_n$, the central map $\mathcal{F}$ of the scheme consists of the two matrices $E_1 = A \cdot B$ and $E_2 = A \cdot C$ and the public key is given as $\mathcal{P} = \mathcal{S} \circ \mathcal{F}$ with an invertible linear transformation $\mathcal{S} : \mathbb{F}^m \to \mathbb{F}^m$. The matrices B and C are chosen to be sparse in the following sense: Each monomial x_i ($i \in \{1, \ldots, n\}$) appears in the linear combinations of every column of B and C exactly once.

Claim. *The above choice of the matrices B and C does not weaken the security of the improved SimpleMatrix encryption scheme.*

It is obvious that for a matrix A containing randomly chosen linear combinations of $x_1, \ldots, x_n$ and matrices B and C chosen as above the components of the central map will be dense, i.e. there exist no systematic blocks of zeros in the central polynomials. Therefore this choice of the matrices B and C does not make our scheme attackable by the UOV Reconciliation attack. Furthermore, our experiments show that our choice of B and C cannot be used by Gröbner basis attacks. However, it remains an open question, if there exist dedicated attacks which can use this structure.

By choosing the matrices B and C in the above way, we can reduce the space needed to store these matrices from $r \cdot (u+v) \cdot n \cdot \log_2(q)$ to $\log_2(r) \cdot (u+v) \cdot n \cdot \log_2(q)$ bits. For the parameters $(\mathrm{GF}(2^{32}), 8, 9, 8, 8, 144, 72)$ this means a reduction of the private key size from 137.25 kB to 114.25 kB by 18%.

8. Efficiency

The most costly step during the decryption process of our scheme is the solution of the linear system given by $W \cdot \bar{E}_1 - B = 0$ and $W \cdot \bar{E}_2 - C = 0$. To solve this system of $r \cdot (u+v)$ linear equations in $s \cdot r + n$ unknowns by Gaussian elimination we need approximately $(s \cdot r + n)^3$ multiplications. Compared to this number, the solution of the quadratic system is easy. Since the system is highly overdetermined, we can do this by the Relinearization algorithm [12], i.e. by solving a linear system of m equations in roughly $(n - r \cdot (u+v-s)) \cdot (n - r \cdot (u+v-s)+1)/2$ variables. The parameters listed in Table 2 are chosen in such a way that the linear system generated by the Relinearization algorithm has a unique solution. Therefore we get the solution of the quadratic system without further testing.

Compared to that, the decryption process of HFE contains the inversion of the univariate polynomial $\bar{\mathcal{F}}(X) = Y$ of large degree D over the extension field $\mathbb{E}$. To do this using Berlekamp's algorithm, one needs about $\mathcal{O}(D^3 + D \cdot \log p^n)$ operations, where p^n is the size of the extension field $\mathbb{E}$. This shows that our scheme is much more efficient than existing multivariate encryption schemes.

To prove our conclusions about the efficiency of our scheme, we created a straightforward C++ implementation of the improved SimpleMatrix encryption scheme. For simplicity reasons we restricted to the case of $\mathrm{GF}(2^8)$ as the underlying field. The scheme runs on a Lenovo Thinkpad with a single AMD Opteron processor with 2.8 GHz and 4 GB RAM. Table 3 shows the results.

Table 3
Running time of our C++ implementation of the improved SimpleMatrix encryption scheme.

parameters $(\mathbb{F}, r, s, u, v, m, n)$	key generation	encryption	decryption	proposed security level (bit)	probability of decryption failure
$\mathrm{GF}(2^8), 8, 11, 12, 12, 264, 128$	239 ms[a] $612 \cdot 10^6$	19 ms $48 \cdot 10^6$	24 ms $60 \cdot 10^6$	80	2^{-32}
$\mathrm{GF}(2^8), 9, 12, 13, 13, 312, 153$	484 ms $1223 \cdot 10^6$	33 ms $83 \cdot 10^6$	39 ms $98 \cdot 10^6$	90	2^{-32}
$\mathrm{GF}(2^8), 8, 13, 14, 14, 364, 180$	794 ms $2033 \cdot 10^6$	53 ms $134 \cdot 10^6$	59 ms $149 \cdot 10^6$	100	2^{-32}

[a] The first number denotes the running time in ms, the second number the number of CPU cycles.

9. Conclusion

In this paper, we presented our results on creating a new multivariate encryption scheme, which is the extension of the original SimpleMatrix encryption scheme. Our scheme allows fast en- and decryption and resists all known attacks against multivariate cryptosystems. Furthermore, our scheme provides a solution for the decryption failure problem of the original SimpleMatrix scheme.

Acknowledgments

We thank the reviewers of Finite Fields and Their Applications for their comments which helped to improve the paper. We thank NSFC for partial support under the grant No. 61472054.

References

[1] D.J. Bernstein, J. Buchmann, E. Dahmen (Eds.), Post Quantum Cryptography, Springer, 2009.

[2] A. Bogdanov, T. Eisenbarth, A. Rupp, C. Wolf, Time-area optimized public-key engines: MQ-cryptosystems as replacement for elliptic curves?, in: CHES 2008, in: Lect. Notes Comput. Sci., vol. 5154, Springer, 2008, pp. 45–61.

[3] A.I.T. Chen, M.-S. Chen, T.-R. Chen, C.-M. Cheng, J. Ding, E.L.-H. Kuo, F.Y.-S. Lee, B.-Y. Yang, SSE implementation of multivariate PKCs on modern x86 CPUs, in: CHES 2009, in: Lect. Notes Comput. Sci., vol. 5747, Springer, 2009, pp. 33–48.

[4] D. Coppersmith, J. Stern, S. Vaudenay, Attacks on the birational signature scheme, in: CRYPTO'94, in: Lect. Notes Comput. Sci., vol. 773, Springer, 1994, pp. 435–443.

[5] J. Ding, J.E. Gower, D.S. Schmidt, Multivariate Public Key Cryptosystems, Springer, 2006.

[6] J. Ding, L. Hu, X. Nie, J. Li, J. Wagner, High Order Linearization Equation (HOLE) attack on multivariate public key cryptosystems, in: PKC 2007, in: Lect. Notes Comput. Sci., vol. 4450, Springer, 2007, pp. 233–248.

[7] J. Ding, D.S. Schmidt, Rainbow, a new multivariate polynomial signature scheme, in: ACNS 2005, in: Lect. Notes Comput. Sci., vol. 3531, Springer, 2005, pp. 164–175.

[8] J. Ding, B.-Y. Yang, C.H.O. Chen, M.-S. Chen, C.-M. Cheng, New differential-algebraic attacks and reparametrization of Rainbow, in: ACNS 2008, in: Lect. Notes Comput. Sci., vol. 5037, Springer, 2008, pp. 242–257.

[9] J.C. Faugere, A new efficient algorithm for computing Gröbner bases (F4), J. Pure Appl. Algebra 139 (1999) 61–88.

[10] L. Goubin, N. Courtois, Cryptanalysis of the TTM cryptosystem, in: ASIACRYPT 2000, in: Lect. Notes Comput. Sci., vol. 1976, Springer, 2000, pp. 44–57.

[11] M.R. Garey, D.S. Johnson, Computers and Intractability: A Guide to the Theory of NP-Completeness, W.H. Freeman and Company, 1979.

[12] A. Kipnis, J. Patarin, L. Goubin, Unbalanced Oil and Vinegar schemes, in: EUROCRYPT 1999, in: Lect. Notes Comput. Sci., vol. 1592, Springer, 1999, pp. 206–222.

[13] D. Kravitz, Digital Signature Algorithm, US patent 5231668, July 1991.

[14] A. Kipnis, A. Shamir, Cryptanalysis of the HFE public key cryptosystem, in: CRYPTO 99, in: Lect. Notes Comput. Sci., vol. 1666, Springer, 1999, pp. 19–30.

[15] T. Matsumoto, H. Imai, Public quadratic polynomial-tuples for efficient signature-verification and message-encryption, in: EUROCRYPT 88, in: Lect. Notes Comput. Sci., vol. 330, Springer, 1988, pp. 419–453.

[16] J. Patarin, N. Courtois, L. Goubin, QUARTZ, 128-bit long digital signatures, in: CTRSA 2001, in: Lect. Notes Comput. Sci., vol. 2020, Springer, 2001, pp. 282–297.

[17] J. Patarin, Cryptanalysis of the Matsumoto and Imai public key scheme of Eurocrypt 88, in: CRYPTO 95, in: Lect. Notes Comput. Sci., vol. 963, Springer, 1995, pp. 248–261.

[18] J. Patarin, Hidden Field Equations (HFE) and Isomorphisms of Polynomials (IP), in: EUROCRYPT'96, in: Lect. Notes Comput. Sci., vol. 1070, Springer, Heidelberg, 1996, pp. 38–48.

[19] R.L. Rivest, A. Shamir, L. Adleman, A method for obtaining digital signatures and public-key cryptosystems, Commun. ACM 21 (2) (1978) 120–126.

[20] P. Shor, Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer, SIAM J. Comput. 26 (5) (1997) 1484–1509.

[21] C. Tao, A. Diene, S. Tang, J. Ding, SimpleMatrix scheme for encryption, in: PQCrypto 2013, in: Lect. Notes Comput. Sci., vol. 7932, Springer, 2013, pp. 231–242.