The Cubic Simple Matrix Encryption Scheme

Jintai Ding[1], Albrecht Petzoldt[2], and Lih-chung Wang[3]

[1] University of Cincinnati, Ohio, USA and Academia Sinica, Taiwan
[2] TU Darmstadt, Germany
[3] National Dong Hwa University, Taiwan
{jintai.ding,lihchungwang}@gmail.com,
apetzoldt@cdc.informatik.tu-darmstadt.de

Abstract. In this paper, we propose an improved version of the Simple Matrix encryption scheme of PQCrypto2013. The main goal of our construction is to build a system with even stronger security claims. By using square matrices with random quadratic polynomials, we can claim that breaking the system using algebraic attacks is at least as hard as solving a set of random quadratic equations. Furthermore, due to the use of random polynomials in the matrix A, Rank attacks against our scheme are not feasible.

Keywords: Multivariate Cryptography, Simple Matrix Encryption Scheme, Provable Security.

1 Introduction

Cryptographic techniques are an essential tool to guarantee the security of communication in modern society. Today, the security of nearly all of the cryptographic schemes 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 [20], DSA 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 [21], 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 like smart cards and RFID chips [2,4]. However, while there exist many practical multivariate signature schemes [9,13,18], the number of efficient and secure multivariate encryption schemes is somewhat limited.

M. Mosca (Ed.): PQCrypto 2014, LNCS 8772, pp. 76–87, 2014.

At PQCrypto 2013, Tao et al. proposed a new MPKC for encryption called Simple Matrix (or ABC) encryption scheme, which resists all known attacks against multivariate schemes. However, decryption errors occur with non negligible probability.

In this paper, we propose an improved version of the ABC scheme. The main goal of our approach is to increase the security of the scheme even further. We achieve this by using square matrices with random quadratic polynomials, by which we obtain a cubic map as the public key. We claim that an algebraic attack on our scheme is at least as hard as solving a random quadratic system of the same size. Furthermore, due to the use of random polynomials in the matrix A, the matrices associated to the central map are of high rank, which prevents the use of Rank attacks against our scheme.

The rest of this paper is organized as follows. In Section 2 we describe the basic ABC encryption scheme as proposed in [22]. Section 3 introduces our cubic version of the ABC scheme. In Section 4 we discuss the security of our scheme, whereas Section 5 proposes concrete parameter sets for the cubic ABC encryption scheme. In Section 6 we describe shortly a technique to decrease the probability of decryption failures. Finally, Section 7 concludes the paper.

2 The Basic ABC Encryption Scheme

In this section we introduce the ABC encryption scheme as proposed by Tao et al. in [22]. Before we come to the description of the scheme itself, we start with a short overview of the main concepts of multivariate cryptography.

2.1 Multivariate Cryptography

The basic objects of multivariate cryptography are systems of multivariate quadratic polynomials (see equation (1)).

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

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) [12].

To build a public key cryptosystem based on 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 $\mathcal{L}_1 : \mathbb{F}^n \to \mathbb{F}^n$ and $\mathcal{L}_2 : \mathbb{F}^m \to \mathbb{F}^m$.
The *public key* is therefore given by $\bar{\mathcal{F}} = \mathcal{L}_2 \circ \mathcal{F} \circ \mathcal{L}_1$.
The *private key* consists of $\mathcal{L}_1$, $\mathcal{F}$ and $\mathcal{L}_2$ 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 Figure 1.

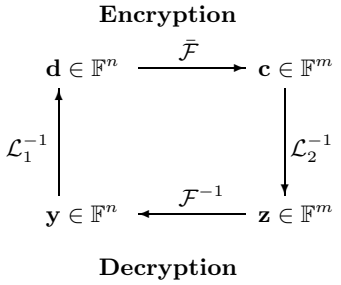

Fig. 1. General workflow of multivariate encryption schemes

Encryption: To encrypt a message $\mathbf{d} \in \mathbb{F}^n$, one simply computes $\mathbf{c} = \bar{\mathcal{F}}(\mathbf{d})$. 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 $\mathbf{z} = \mathcal{L}_2^{-1}(\mathbf{c})$, $\mathbf{y} = \mathcal{F}^{-1}(\mathbf{z})$ and $\mathbf{d} = \mathcal{L}_1^{-1}(\mathbf{y})$. $\mathbf{d} \in \mathbb{F}^n$ is the plaintext corresponding to the ciphertext $\mathbf{c}$.

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

An overview of existing multivariate schemes can be found in [8].

2.2 The ABC Encryption Scheme of [22]

The original Simple Matrix encryption scheme as proposed by Tao et al. 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 $m = 2 \cdot n$ and define three matrices A, B and C of the form

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

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

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 $\bar{\mathcal{F}} = \mathcal{L}_2 \circ \mathcal{F} \circ \mathcal{L}_1 : \mathbb{F}^n \to \mathbb{F}^m$ with two randomly chosen invertible linear maps $\mathcal{L}_2 : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{L}_1 : \mathbb{F}^n \to \mathbb{F}^n$, the *private key* consists of the matrices B and C and the linear maps $\mathcal{L}_1$ and $\mathcal{L}_2$.

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

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

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

$$\bar{E}_1 = \begin{pmatrix} z_1 & \dots & z_s \\ \vdots & & \vdots \\ z_{(s-1)\cdot s+1} & \dots & z_n \end{pmatrix}, \; \bar{E}_2 = \begin{pmatrix} z_{n+1} & \dots & z_{n+s} \\ \vdots & & \vdots \\ z_{n+(s-1)\cdot s+1} & \dots & z_m \end{pmatrix}.$$

2. In the second step one has to find a vector $\mathbf{y} = (y_1, \dots, y_n)$ such that $\mathcal{F}(\mathbf{y}) = \mathbf{z}$. 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 $y_1, \dots, y_n$.
   - 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 $\bar{A} = A(\mathbf{y})$ 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 $w_1, \dots, w_n$ and therefore gets m linear equations in the m variables $w_1, \dots, w_n, y_1, \dots, y_n$.
   - 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 $\mathbf{d} = \mathcal{L}_1^{-1}(y_1, \dots, y_n)$.

The probability of a decryption failure occurring in the second step is about $\frac{1}{q}$.

It might happen that the linear systems in the second step of the decryption process have multiple solutions $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(\ell)}$. In this case one has to perform the third step 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 $\mathbf{c}$.

3 The New Cubic Encryption Scheme

The new cubic Simple Matrix encryption scheme 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 $m = 2 \cdot n$ and define three matrices A, B and C of the form

$$A = \begin{pmatrix} p_1 & \cdots & p_s \\ \vdots & & \vdots \\ p_{(s-1)\cdot s+1} & \cdots & p_n \end{pmatrix}, \; B = \begin{pmatrix} b_1 & \cdots & b_s \\ \vdots & & \vdots \\ b_{(s-1)\cdot s+1} & \cdots & b_n \end{pmatrix}, \; C = \begin{pmatrix} c_1 & \cdots & c_s \\ \vdots & & \vdots \\ c_{(s-1)\cdot s+1} & \cdots & c_n \end{pmatrix}.$$

Here, $p_1, \ldots, p_n$ are random quadratic polynomials, whereas $b_1, \ldots, b_s$ 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 $\bar{\mathcal{F}} = \mathcal{L}_2 \circ \mathcal{F} \circ \mathcal{L}_1 : \mathbb{F}^n \to \mathbb{F}^m$ with two randomly chosen invertible linear maps $\mathcal{L}_2 : \mathbb{F}^m \to \mathbb{F}^m$ and $\mathcal{L}_1 : \mathbb{F}^n \to \mathbb{F}^n$, the *private key* consists of the matrices B and C and the linear maps $\mathcal{L}_1$ and $\mathcal{L}_2$.

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

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

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

$$\bar{E}_1 = \begin{pmatrix} z_1 & \cdots & z_s \\ \vdots & & \vdots \\ z_{(s-1)\cdot s+1} & \cdots & z_n \end{pmatrix}, \; \bar{E}_2 = \begin{pmatrix} z_{n+1} & \cdots & z_{n+s} \\ \vdots & & \vdots \\ z_{n+(s-1)\cdot s+1} & \cdots & z_m \end{pmatrix}.$$

2. In the second step one has to find a vector $\mathbf{y} = (y_1, \ldots, y_n)$ such that $\mathcal{F}(\mathbf{y}) = \mathbf{z}$. 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 $y_1, \ldots, y_n$.
   - 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 $\bar{A} = A(\mathbf{y})$ 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 $w_1, \ldots, w_n, y_1, \ldots, y_n$.
   - 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 $\mathbf{d} = \mathcal{L}_1^{-1}(y_1, \ldots, y_n)$.

The probability of a decryption failure occurring in the second step is about $\frac{1}{q}$.

It might happen that the linear systems in the second step have multiple solutions $\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(\ell)}$. 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 Security Analysis

4.1 Rank Attacks

Rank attacks are one of the major threats against multivariate encryption schemes. There are two different versions of this attack. The first one is called the MinRank attack or LowRank attack as proposed by Goubin et al. in [11]. The other one is called the HighRank Attack [5].

The goal of the MinRank attack is to find a linear combination of the components of the public key of minimal rank r. In the context of e.g. HFE such a polynomial of low rank corresponds to a central polynomial. By finding those linear combinations of low rank an attacker can recover the linear map $\mathcal{L}_2$ and therefore the secret key of the scheme.

In the High Rank Attack, the attacker tries to find linear combinations corresponding to variables which appear in the central polynomials the smallest number of times. In a scheme like Rainbow these are the oil variables of the last layer. By repeating this attack for the other layers, the attacker can recover the linear map $\mathcal{L}_1$ and therefore the secret key of the scheme.

However, in the case of the cubic Simple Matrix encryption scheme, the elements of the matrix A are randomly chosen multivariate quadratic polynomials. Therefore, their rank is close to n and all variables appear in each of the central polynomials approximately the same number of times. This shows that rank attacks can not be used to attack the cubic Simple Matrix encryption scheme.

4.2 Algebraic Attacks

In a direct attack (message recovery attack) the attacker tries to solve the public system $\mathcal{F}(\mathbf{d}) = \mathbf{c}$ for the plaintext $\mathbf{d}$. To achieve this, the attacker can use either a Gröbner Basis method such as F_4[10] or a system solving algorithm like XL or one of its variants like mutant XL [6,7,17,16].

When attacking our scheme, the attacker is faced with a system of $m = 2n$ multivariate cubic polynomials in n variables. As described in the section above, this system was obtained by multiplying a matrix A containing randomly chosen multivariate quadratic polynomials with matrices B and C containing linear ones (when neglecting the linear transformations $\mathcal{L}_1$ and $\mathcal{L}_2$). We make the following claim:

Claim: Solving the cubic public system of our scheme is asymptotically at least as hard as solving a multivariate quadratic system with randomly chosen coefficients.

To justify this claim, let us assume that an attacker wants to solve the equation $E_2(\mathbf{x}) = \mathbf{y}$, where $E_2 = A \cdot C$ and $\mathbf{y}$ is some matrix in $\mathbb{F}^{s \times s}$. Let us further assume that an oracle $\mathcal{O}$ gives the attacker the values of the elements of C (without revealing the inner structure of this matrix), i.e. the oracle gives him a matrix $\bar{C} \in \mathbb{F}^{s \times s}$ with $\bar{C} = C(\mathbf{x})$. So the attacker obtains a system of linear combinations in the elements of the matrix A. By solving this system by Gaussian elimination, the attacker finally gets a system $A(\mathbf{x}) = \mathbf{y} \cdot \bar{C}^{-1}$. But to get the values of $(x_1, \ldots, x_n)$, the attacker still has to solve a system of multivariate quadratic equations with randomly chosen coefficients.

A much more interesting heuristic argument goes as follows.

Let us denote the polynomial entries of the matrix A by $A_{ij}(\mathbf{x})$ and similarly the polynomial entries of E_1 and E_2 by $E_{1,ij}(\mathbf{x})$ and $E_{2,ij}(\mathbf{x})$. And we denote the entries of B and C by $B_{ij}(\mathbf{x})$ and $C_{ij}(\mathbf{x})$ respectively. Clearly we have that

$$E_{1,ij}(\mathbf{x}) = \sum_{l=1}^{s} A_{il}(\mathbf{x}) \cdot B_{lj}(\mathbf{x}),$$

$$E_{2,ij}(\mathbf{x}) = \sum_{l=1}^{s} A_{il}(\mathbf{x}) \cdot C_{lj}(\mathbf{x}).$$

In the case of quadratic systems, it is a common assumption that the complexity of solving the system is actually determined by the structure of the ideal generated by the homogeneous part of highest degree, namely the degree 2 part of the polynomials.

In our case this means that the complexity of solving the system

$$A_{ij}(\mathbf{x}) = D_{ij},$$

is actually determined by the structure of the ideal generated by the homogeneous polynomials $\bar{A}_{ij}(\mathbf{x})$, which are the quadratic part of $A_{ij}(\mathbf{x})$. We call this ideal I_A.

Now let us look at the system $E_{1,ij}(\mathbf{x}) = D_{1,ij}$, and $E_{2,ij}(\mathbf{x}) = D_{2,ij}$. In this case the complexity should be dominated by the structure of the homogeneous part of degree 3, which is given by

$$\bar{E}_{1,ij}(\mathbf{x}) = \sum_{l=1}^{s} \bar{A}_{il}(\mathbf{x}) \cdot \bar{B}_{lj}(\mathbf{x}) \text{ and}$$

$$\bar{E}_{2,ij}(\mathbf{x}) = \sum_{l=1}^{s} \bar{A}_{il}(\mathbf{x}) \cdot \bar{C}_{lj}(\mathbf{x}),$$

where $\bar{B}_{ij}$ and $\bar{C}_{ij}$ are the homogeneous linear parts of B_{ij} and C_{ij} respectively. We call this ideal I_E. If we now look that the generators of this ideal, we immediately reach the conclusion that

$$I_E \subset I_A.$$

Furthermore, the generators of I_E are nothing but elements in the space spanned by the elements generated in the first step of the XL algorithm if applied to I_A, since $\bar{B}_{ij}(\mathbf{x})$ and $\bar{C}_{ij}(\mathbf{x})$ are nothing but linear functions. From this perspective, we therefore speculate that in general or precisely asymptotically (when s is too small it might be different), the complexity of solving the public systems of the cubic Simple Matrix encryption scheme should be harder or at least as hard as solving a quadratic system with randomly chosen coefficients of size $n \times n$.

This heuristic analysis is very speculative, however it is very exciting in the sense that it actually hints that maybe we can derive a certain form of provable security for our new system, which is something we have never seen before.

Additionally to these theoretical considerations, we carried out a number of experiments with MAGMA, which contains an efficient implementation of Faugeres F_4 algorithm [10]. For this, we created, for different parameter sets, the public system of both the cubic Simple Matrix encryption scheme and the original ABC scheme of [22] and solved these systems using the MAGMA command `Variety`. We repeated each of these experiments ten times.

Table 1. Direct attack against the cubic Simple Matrix encryption scheme

		GF(2^8)				GF(2^{16})			
	(s, m, n)	$(2, 4, 8)$	$(3, 9, 18)$	$(4, 16, 32)$	$(5, 25, 50)$	$(2, 4, 8)$	$(3, 9, 18)$	$(4, 16, 32)$	$(5, 25, 50)$
our scheme	d_{reg}	5	6	7	-	5	6	7	-
	time(s)	0.8	15.4	-	-	1.2	23.7	-	-
	memory(MB)	5.2	18.3	ooM [1]	-	8.4	27.1	ooM [1]	-
ABC scheme of [22]	d_{reg}	-	4	5	6	-	4	5	6
	time(s)	-	0.02	3.5	17,588	-	0.1	5.7	23,264
	memory(MB)	-	3.4	8.1	1,112	-	7.4	23.1	3,214

[1] out of memory

As we see from the table, for the same value of s the degree of regularity is at least higher by two than that of the original Simple Matrix encryption scheme. Therefore, to obtain the same security level, we can decrease the value of s by 2 (compared to [22]).

Here we would like to point out that, due to the fact that we can only perform experiments for very small s, we can not really say anything precise about our speculations.

5 Parameter Proposals

Based on our security analysis presented in the previous section, we propose the following parameters for our cubic version of the Simple Matrix encryption scheme. For the fields GF(2^8) and GF(2^{16}), to be on the conservative side, we suggest

- $s = 7$ for a security level of 80 bit and
- $s = 8$ for a security level of 100 bit

These parameter proposals are obtained by the following analysis:

As Table 1 shows, the degree of regularity of solving the public system increases linearly with s. We can therefore assume that for $s = 7$ the degree of regularity is greater or equal to 10, while for $s = 8$ it is given by 11. We can therefore (for $s = 7$) estimate the number of homogeneous monomials of highest degree in the solving step of F_4 by

$$T = \binom{n + d_{\text{reg}}}{d_{\text{reg}}} \geq 2^{35.8}.$$

The number of non-zero monomials in every polynomial is given by

$$\tau = \binom{n+3}{3} \geq 2^{14.4}.$$

Therefore we can estimate the complexity of a direct attack against the cubic Simple Matrix Encryption scheme by

$$\text{Complexity}_{\text{direct attack}}(s = 7) \geq 3 \cdot \tau \cdot T^2 \geq 2^{88}. \tag{2}$$

For $s = 8$ we get $T \geq 2^{42.2}$, $\tau \geq 2^{15.5}$ and therefore

$$\text{Complexity}_{\text{direct attack}}(s = 8) \geq 3 \cdot \tau \cdot T^2 \geq 2^{102}. \tag{3}$$

Table 2 shows for our 4 parameter sets key sizes of the cubic Simple Matrix encryption scheme as well as the probability of a failure occurring during the decryption process.

Table 2. Parameters and key sizes of the cubic Simple Matrix encryption scheme

security level (bit)	parameters $(\mathbb{F}, s, n, m)$	input size (bit)	output size (bit)	public key size (kB)	private key size(kB)	probability of decryption failure
80	(GF(2^8),7,49,98)	392	784	2,115	72.7	2^{-8}
	(GF(2^{16},7,49,98)	784	1,568	4,230	145.4	2^{-16}
100	($GF(2^8)$,8,64,128)	512	1,024	5,988	154	2^{-8}
	(GF(2^{16},8,64,128)	1,024	2,048	11,976	308	2^{-16}

Here, we would like to further speculate actually that even for the case of $s = 5$ and $s = 6$, the scheme might provide a good security level for practical applications. But this needs much better support evidence, which we still do not have.

6 Decreasing the Probability of Decryption Failures

As Table 2 shows, the probability of failures occuring during the decryption process of our scheme is non negligible. To decrease this probability, we can use the technique presented in [23]. The basic idea of this is to use non square matrices for A, B and C. In particular, A is chosen to be an $r \times s$ ($r < s$) matrix containing random quadratic polynomials, while the matrices B and C (containing linear combinations of $x_1, \ldots, x_n$) are of size $s \times u$. Decryption remains possible, as long as the rank of the matrix A is at least r. The probability of this is given by

$$\Pr(\text{Rank}(A) \geq r) = 1 - \left(1 - \frac{1}{q^s}\right) \cdot \left(1 - \frac{1}{q^{s-1}}\right) \cdot \ldots \cdot \left(1 - \frac{1}{q^{s-r+1}}\right) \approx \frac{1}{q^{s-r+1}}.$$

By choosing the parameters r and s of the scheme in an appropriate way, it is possible to decrease the probability of decryption failures arbitrarily.

In [23] it was shown that by this strategy the security of the scheme against known attacks is not weakened. However, as it comes to provable security, we do not exactly know what happens. In this field, there has still much work to be done.

However, there are other ways to reduce the probability of decryption failures. For example, we can simply encrypt the message twice. In the second run, we encode the message with a public invertible affine transformation over the ring $\mathbb{Z}_{256}$ (integers mod 256) and then encrypt the encoded message with our scheme. Since an affine transformation over the ring $\mathbb{Z}_{256}$ is algebraically complicated with respect to the Galois field GF(256), we can not join these two encryptions as a larger low-degree algebraic system. If necessary, one can even encrypt the message several times. We will demonstrate the ways to reduce decryption failures in our further work.

7 Conclusion and Future Work

In this paper we proposed a cubic version of the Simple Matrix encryption scheme of PQCrypto 2013 [22]. By using a matrix A whose elements are randomly chosen multivariate quadratic polynomials, we increase the security of the original Simple Matrix scheme even further. Our construction completely eliminates the possibility of Rank attacks against our scheme. Furthermore, we speculate that breaking our scheme using direct attacks is as least as hard as solving a quadratic system with randomly chosen coefficients. Future work includes decreasing the probability of decryption failures and a formal proof of our security claim.

Acknowledgements. The first author was partially supported by the CAS/SAFEA International Partnership Program for Creative Research Teams.

References

1. Bernstein, D.J., Buchmann, J., Dahmen, E. (eds.): Post Quantum Cryptography. Springer (2009)
2. Bogdanov, A., Eisenbarth, T., Rupp, A., Wolf, C.: Time-area optimized public-key engines: MQ-cryptosystems as replacement for elliptic curves? In: Oswald, E., Rohatgi, P. (eds.) CHES 2008. LNCS, vol. 5154, pp. 45–61. Springer, Heidelberg (2008)
3. Bardet, M., Faugere, J.-C., Salvy, B.: On the complexity of Gröbner basis computation of semi-regular overdetermined algebraic equations. In: ICPSS, pp. 71–75 (2004)
4. Chen, A.I.T., Chen, M.-S., Chen, T.-R., Cheng, C.-M., Ding, J., Kuo, E.L.-H., Lee, F.Y.-S., Yang, B.-Y.: SSE implementation of multivariate pkcs on modern x86 cpus. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 33–48. Springer, Heidelberg (2009)
5. Coppersmith, D., Stern, J., Vaudenay, S.: Attacks on the Birational Permutation Signature Schemes. In: Stinson, D.R. (ed.) CRYPTO 1993. LNCS, vol. 773, pp. 435–443. Springer, Heidelberg (1994)

6. Courtois, N.T., Klimov, A., Patarin, J., Shamir, A.: Efficient algorithms for solving overdefined systems of multivariate polynomial equations. In: Preneel, B. (ed.) EUROCRYPT 2000. LNCS, vol. 1807, pp. 392–407. Springer, Heidelberg (2000)
7. Ding, J., Buchmann, J., Mohamed, M.S.E., Mohamed, W.S.A.E., Weinmann, R.-P.: Mutant XL. Talk at the First International Conference on Symbolic Computation and Cryptography (SCC 2008), Beijing (2008)
8. Ding, J., Gower, J.E., Schmidt, D.S.: Multivariate Public Key Cryptosystems. Springer (2006)
9. Ding, J., Schmidt, D.: Rainbow, a new multivariable polynomial signature scheme. In: Ioannidis, J., Keromytis, A.D., Yung, M. (eds.) ACNS 2005. LNCS, vol. 3531, pp. 164–175. Springer, Heidelberg (2005)
10. Faugere, J.C.: A new efficient algorithm for computing Gröbner bases (F4). Journal of Pure and Applied Algebra 139, 61–88 (1999)
11. Goubin, L., Courtois, N.: Cryptanalysis of the TTM Cryptosystem. In: Okamoto, T. (ed.) ASIACRYPT 2000. LNCS, vol. 1976, pp. 44–57. Springer, Heidelberg (2000)
12. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W.H. Freeman and Company (1979)
13. Kipnis, A., Patarin, J., Goubin, L.: Unbalanced Oil and Vinegar Signature Schemes. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 206–222. Springer, Heidelberg (1999)
14. Kipnis, A., Shamir, A.: Cryptanalysis of the HFE Public Key Cryptosystem by Relinearization. In: Wiener, M. (ed.) CRYPTO 1999. LNCS, vol. 1666, pp. 19–30. Springer, Heidelberg (1999)
15. Matsumoto, T., Imai, H.: Public quadratic polynomial-tuples for efficient signature-verification and message-encryption. In: Günther, C.G. (ed.) EUROCRYPT 1988. LNCS, vol. 330, pp. 419–453. Springer, Heidelberg (1988)
16. Mohamed, M.S.E., Cabarcas, D., Ding, J., Buchmann, J., Bulygin, S.: MXL_3: An efficient algorithm for computing Gröbner bases of zero-dimensional ideals. In: Lee, D., Hong, S. (eds.) ICISC 2009. LNCS, vol. 5984, pp. 87–100. Springer, Heidelberg (2010)
17. Mohamed, M.S.E., Mohamed, W.S.A.E., Ding, J., Buchmann, J.: MXL2: Solving polynomial equations over GF(2) using an improved mutant strategy. In: Buchmann, J., Ding, J. (eds.) PQCrypto 2008. LNCS, vol. 5299, pp. 203–215. Springer, Heidelberg (2008)
18. Patarin, J., Courtois, N., Goubin, L.: QUARTZ, 128-Bit Long Digital Signatures. In: Naccache, D. (ed.) CT-RSA 2001. LNCS, vol. 2020, pp. 282–297. Springer, Heidelberg (2001)
19. Patarin, J.: Cryptanalysis of the Matsumoto and Imai Public Key Scheme of Eurocrypt '88. In: Coppersmith, D. (ed.) CRYPTO 1995. LNCS, vol. 963, pp. 248–261. Springer, Heidelberg (1995)
20. Rivest, R.L., Shamir, A., Adleman, L.: A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. Commun. ACM 21(2), 120–126 (1978)
21. Shor, P.: Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. SIAM J. Comput. 26(5), 1484–1509
22. Tao, C., Diene, A., Tang, S., Ding, J.: Simple Matrix Scheme for Encryption. In: Gaborit, P. (ed.) PQCrypto 2013. LNCS, vol. 7932, pp. 231–242. Springer, Heidelberg (2013)
23. Tao, C., Petzoldt, A., Ding, J.: SimpleMatrix - An MPKC for Encryption (to appear)