GANITA,Vol.74(2), 2024, 93-103

A NEW CENTRAL MAP FOR MULTIVARIATE ENCRYPTION SCHEME

SUMAN LATA VERMA ✉ **and HEMLAL SAHU**

Abstract

The Central map plays essential role in design multivariate cryptosystems. This paper propose a new central map based on simple matrix encryption to achieve more security. The proposed scheme's significant advantage is using present square matrices with random polynomials. we claim that breaking the system using algebraic attack is at least as complex as solving a set of cubic equations [**?**]. due to use of random polynomials in matrix A, Rank attacks against our scheme are not feasible.

2010 *Mathematics subject classification:* primary 94A60; secondary 14G50.

Keywords and phrases: Post quantum cryptography , Encryption Schemes, Multivariate polynomials..

1. Introduction

After the invention of RSA much research has been done for developing new public key Cryptography, Which we can use for different purposes because traditional public keys are broken by Shor's [2] algorithm for factoring and computing discrete logarithms with quantum computers, For the large quantum computers this algorithm is not well suited. much of the research has been devoted to constructing a new system for use in small devices with limited computing resources and also free from attacks, the post quantum Era is started and to look such a system is based on Multivariate quadratic polynomials. Multivariate cryptography is the study of public key cryptosystems, Public key cryptosystems are depends on the class of the "trapdoor one way function" in the study of multivariate cryptography this trapdoor one way function takes the form of a multivariate polynomial map over a finite field. The PQC family includes multivariate public key cryptosystems (MPKC).They might make a suitable candidate for PQC if they are well-designed. A system of multivariate polynomials, typically quadratic, over a finite field makes up the public key of an MPKC. Understanding that a collection of multivariate polynomial equations over a finite field may be solved is the foundation for the security of MPKCs.

Matsumoto and Imai gave the first multivariate quadratic scheme [3] (MI or C*).They suggested a set of multivariate polynomial equations over a finite field, However this scheme has broken by patarin's linear method [4]. For efficiency purpose the rank of the quadratic form restrict (in most cases) to the central map of C* is two

therefore secrete key can be also recovered by MinRank Attack. Patarin had the idea by using a new central map to construct a new encryption scheme named Hidden Field Equations (HFE) is originated not only for define the scheme against the Linearization Equations attack but also increased its security against Rank attacks

Due to the special properties of multivariate cryptography a new encryption scheme has invented depend on the matrix multiplication (Simple Matrix) scheme for new central map with high rank. Simple Matrix [5] encryption technique is a novel MPKC for encryption that Tao et al. proposed at PQCrypto 2013 and which depends off every known attack on multivariate methods. Decryption mistakes do, however, happen occasionally. Ding et al. proposed an improved version of the ABC scheme (cubic matrix simple encryption scheme). The Cubic Simple Matrix [1] Encryption Scheme claims to have higher security than the Simple Matrix Encryption Scheme. The scheme uses square matrices with random quadratic polynomials to build a system with even stronger security claims. The Cubic Simple Matrix Encryption Scheme makes the security claim that it is at least as difficult to break the system using algebraic attacks as it is to solve a collection of randomly generated quadratic equations. The problem of decryption failures occurring in the Simple Matrix scheme has been raised. However, while these approaches could reduce the probability of decryption failures occurring, a general solution to the problem is still missing. In this paper, we propose an improved version of the cubic ABC scheme. The main goal of our approach is to increase the security of the scheme even further. We achieve this by generating a new central map. the public key associated with three matrices as ABC where elements of A are the linear monomials of the multivariate polynomial ring. and Elements of B and C are randomly chosen linear combination.

The rest of this paper is organized as follows. In Section 2 we describe the basic ABC and Cubic ABC proposed In [5], [1]. Section 3 introduced our new central map. In section 4 we discuss the security of our scheme, whereas Section 5 propose concrete parameter sets for new scheme. while Section 6 discusses the efficiency Finally Section 7 conclude the paper.

2. Description of Simple Matrix Encryption Scheme

This section briefly explains the definition, notation, and properties of The basic ABC and Cubic Simple matrix encryption schemes. We begin with a brief explanation of the key ideas in multivariate cryptography before describing the scheme itself,

2.1. Multivariate Cryptography: Systems of multivariate quadratic polynomials are the fundamental components of multivariate cryptography.

$$p^1(x_1.....x_n) = \sum_{i=1}^{n}\sum_{j=1}^{n} p_i^1 j.x_i x_j + \sum_{i=1}^{n} p_i^1.x_i + p_0^1$$

$$p^2(x_1.....x_n) = \sum_{i=1}^{n}\sum_{j=1}^{n} p_i^2 j.x_i x_j + \sum_{i=1}^{n} p_i^2.x_i + p_0^2$$

$$p^m(x_1.....x_n) = \sum_{i=1}^{n}\sum_{j=1}^{n} p_i^m j.x_i x_j + \sum_{i=1}^{n} p_i^m.x_i + p_0^m$$

The security of Multivariate schemes is based on the

Problem MQ: Given m multivariate quadratic polynomials $p^1(x).....p^m(x)$ find a vector $\overline{x} = (\overline{x}_1, ..., \overline{x}_n)$ such that $p^1(\overline{x}) = ...p^m(\overline{x}) = 0$. the MQ problem (for $m \approx n$) is proven to be NP hard even for quadratic polynomials over the field GF (2) [6]
The basic idea behind multivariate public key, Choose $F : F^n \to F^m$ (central map).To hide the structure of F in the public key one composes it with two invertible affine or linear maps $L_1 : F^n \to F^n$ and $L_2 : F^m \to F^m$.
The public key is therefore given by $\overline{F} = L_2 o F o L_1$.
The private key consists of L_1, F, L_2 and therefore allows to invert the public key.

In this paper we concentrate on Multivariate encryption schemes.

$d \in F^n \overline{F} c \in F^m$

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

Decryption: To decrypt the ciphertext $c \in F^m$, one computes recursively $z = {L_2}^{-1}$(c), $y = F^{-1}$(z) and $d = L_1^{-1}$(y).$d \in F^n$ is the plaintext corresponding to the ciphertext c.
Since, for multivariate encryption schemes, we have $m \geq n$,the preimage of the vector z under the central map F and therefore the decrypted plaintext is unique.

An overview of existing multivariate schemes can be found in [7].

2.2. ABC scheme: The initial straightforward matrix encryption method put out by Toa et al. can be summarized as follows.
Key Generation: Let F be a finite field with q elements. For a parameter $s \in N$, we set $n = s^2$ and $m = 2n$ and define three matrices A, B and C of the form:

$$\text{A} = \begin{pmatrix} x_1 & \cdots x_s \\ x_{s^2+1} & \cdots x_{2s} \\ x_{(s-1)s+1} & \cdots x_n \end{pmatrix}, \text{B} = \begin{pmatrix} b_1 & \cdots b_s \\ b_{s^2+1} & \cdots b_{2s} \\ b_{(s-1)s+1} & \cdots b_n, \end{pmatrix}, \text{C} = \begin{pmatrix} c_1 & \cdots c_s \\ c_{s^2+1} & \cdots c_{2s} \\ c_{(s-1)s+1} & \cdots C_n \end{pmatrix}$$

Here $x_1.....x_n$ are the linear monomials of the multivariate polynomial ring $F[x_1.....x_n]$, whereas $b_1.....b_n$ and $c_1.....c_n$ are randomly chosen linear combination of $x_1.....x_n$ computes $E_1 = A.B$ and $E_2 = A.C$
The central map 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 $P = SoFoT : F^n \to F^m$ with two randomly chosen invertible linear maps $S : F^m \to F^m$ and $T : F^n \to F^n$ the private key consists of the matrices B and C and the linear maps S and T.

Encryption: To encrypt a message $d \in F^n$, one simply computes $c = p(d) \in F^m$

Decryption: To decrypt a message $c \in F^m$, one has to perform the following steps

1. Compute $x = s^{-1}(c)$ the elements of the vector $x \in F^m$ are written into matrices $\overline{E_1}$ and $\overline{E_2}$ as follows-
$$\overline{E_1} = \begin{pmatrix} Z_1 & \cdots Z_s \\ Z_{s^2+1} & \cdots \quad {}_{2s} \\ z_{(s-1)s+1} & \cdots \quad z_n \end{pmatrix}, \overline{E_2} = \begin{pmatrix} Z_n+1 & \cdots Z_n+2 \\ Z_{n+s^2+1} & \cdots Z_{n+2s} \\ z_{n+(s-1)s+1} & \cdots z_m \end{pmatrix}$$
2. In the second step $E_1 = A.B$ and $E_2 = A.C$ four cases arises -

   - If E_1 is invertible, then $B.E_1^{-1}.E_2 = \mathrm{C}$. We have n linear equations with n unknowns $x_1, ..., x_n$.
   - If E_2 is invertible, but E_1 is not invertible, then $C.{E_2}^{-1}E_1 = B$. We also have n linear equations with n unknowns $x_1, ..., x_n$
   - If both E_1 and E_2 are not invertible but A is invertible, then $A^{-1}E_1 = B$, $A^{-1}E_2 = C$ We interpret the elements of A^{-1} as the new variables W_i and we end up with $m = 2n$ linear equations in m unknowns. Then we eliminate the new variables to derive n linear equations in the x_i.
   - If none of invertible there occurs a decryption failure. Finally, one computes the plaintext by d. The probability of a decryption failure occurring in the second step is about $1/q$, where q is the cardinality of the underlying field F. It might happen that the linear systems in the second step of the decryption process have multiple solutions $y_1, ..., y_l$. In this case one has to perform the third step for each of these solutions to get a set of possible plaintexts $d_1, ..., d_l$.By encrypting these plaintexts one can test which of them corresponds to the given ciphertext c.

2.3. Description of Cubic Simple Matrix Encryption Scheme: The Cubic Simple Matrix Encryption scheme is as follows :
Key Generation : Let F be a finite field with q elements. For a parameter $s \in N$, we set $n = s^2$ and $m = 2n$ and define three matrices A, B and C of the form:
$$\mathrm{A} = \begin{pmatrix} p_1 & \cdots p_s \\ p_{s^2+1} & \cdots p_{2s} \\ p_{(s-1)s+1} & \cdots p_n \end{pmatrix}, \mathrm{B} = \begin{pmatrix} b_1 & \cdots b_s \\ b_{s^2+1} & \cdots b_{2s} \\ b_{(s-1)s+1} & \cdots b_n \end{pmatrix}, \mathrm{C} \begin{pmatrix} c_1 & \cdots c_s \\ c_{s^2+1} & \cdots c_{2s} \\ c_{(s-1)s+1} & \cdots C_n \end{pmatrix}$$
Here $p_1 \ldots\ldots\ldots\ldots p_n$ are randomly quadratic polynomials, whereas $b_1 \ldots\ldots\ldots b_n$ and $c_1 \ldots c_n$ are randomly chosen linear combination of $x_1 \ldots\ldots\ldots\ldots\ldots\ldots x_n$ computes $E_1 = A.B$ and $E_2 = A.C$ The central map 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 $P = SoFoT : F^n \to F^m$ with two randomly chosen invertible linear maps $S : F^m \to F^m$ and $T : F^n \to F^n$ the private key consists of the matrices B and C and the linear maps S and T.

Encryption: To encrypt a message $d \in F^n$, one simply computes $c = p(d) \in F^m$

Decryption: To decrypt a message $c \in F^m$, one has to perform the following three steps

1. Compute $x = s^{-1}(c)$ the elements of the vector $x \in F^m$ are written into matrices $\overline{E_1}$ and $\overline{E_2}$ as follows-
$$\overline{E_1} = \begin{pmatrix} Z_1 & \cdots Z_s \\ Z_{s^2+1} & \cdots Z_{2s} \\ z_{(s-1)s+1} & \cdots z_n \end{pmatrix}, \overline{E_2} = \begin{pmatrix} Z_n+1 & \cdots Z_n+2 \\ Z_{n+s^2+1} & \cdots Z_{n+2s} \\ z_{n+(s-1)s+1} & \cdots z_m \end{pmatrix}$$
2. In the second step $E_1 = A.B$ and $E_2 = A.C$ four cases arises -

   - If E_1 is invertible, then $B.E_1^{-1}.E_2 = \mathrm{C}$. We have n linear equations with n unknowns $x_1, ..., x_n$.
   - If E_2 is invertible, but E_1 is not invertible, then $C.E_2{}^{-1}E_1 = B$. We also have n linear equations with n unknowns $x_1, ..., x_n$
   - If both E_1 and E_2 are not invertible but A is invertible, then $A^{-1}E_1 = B$, $A^{-1}E_2 = C$ We interpret the elements of A^{-1} as the new variables W_i and we end up with $m = 2n$ linear equations in m unknowns. Then we eliminate the new variables to derive n linear equations in the x_i.
   - If none of invertible there occurs a decryption failure. Finally, one computes the plaintext by d. The probability of a decryption failure occurring in the second step is about $1/q$, where q is the cardinality of the underlying field F. It might happen that the linear systems in the second step of the decryption process have multiple solutions $y_1, ..., y_l$. In this case one has to perform the third step for each of these solutions to get a set of possible plaintexts $d_1, ..., d_l$.By encrypting these plaintexts one can test which of them corresponds to the given ciphertext c.

3. Description of Improved Cubic Simple Matrix Encryption Scheme

Encryption scheme is as follows :
Key Generation : Let F be a finite field with q elements. For a parameter $s \in N$, we set $n = s^2$ and $m = 2n$ and define three matrices A, B and C of the form:
$$\mathrm{A} = \begin{pmatrix} x_1 & \cdots x_s \\ x_{s^2+1} & \cdots x_{2s} \\ x_{(s-1)s+1} & \cdots x_n \end{pmatrix}, \mathrm{B} = \begin{pmatrix} b_1 & \cdots b_s \\ b_{s^2+1} & \cdots b_{2s} \\ b_{(s-1)s+1} & \cdots b_n \end{pmatrix}, \mathrm{C} = \begin{pmatrix} c_1 & \cdots c_s \\ c_{s^2+1} & \cdots c_{2s} \\ c_{(s-1)s+1} & \cdots C_n \end{pmatrix}$$
Here $x_1 \ldots\ldots\ldots x_n$ are linear polynomials, whereas $b_1 \ldots\ldots\ldots b_n$ and $c_1 \ldots\ldots c_n$ are randomly choosen linear combination of $x_1 \ldots\ldots\ldots x_n$ computes $E_1 = AC.B^{-1}$ and $E_2 = AB.C^{-1}$ The central map 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 $P = SoFoT : F^n \to F^m$ with two randomly chosen invertible linear maps $S : F^m \to F^m$ and $T : F^n \to F^n$ the private key consists of the matrices B and C and the linear maps S and T.

Encryption: To encrypt a message $d \in F^n$, one simply computes $c = p(d) \in F^m$

Decryption: To decrypt a message $c \in F^m$, one has to perform the following three steps:

1. Compute $x = s^{-1}(c)$ the elements of the vector $x \in F^m$ are written into matrices $\overline{E_1}$ and $\overline{E_2}$ as follows-
$$\overline{E_1} = \begin{pmatrix} Z_1 & \cdots Z_s \\ Z_{s^2+1} & \cdots Z_{2s} \\ z_{(s-1)s+1} & \cdots z_n \end{pmatrix}, \overline{E_2} = \begin{pmatrix} Z_{n+1} & \cdots Z_n + 2 \\ Z_{n+s^2+1} & \cdots Z_{n+2s} \\ z_{n+(s-1)s+1} & \cdots z_m \end{pmatrix}$$

2. In the second step $E_1 = AC.B^{-1}$ and $E_2 = AB.C^{-1}$ four cases arises -

   - If E_1 is invertible, then $B^2 = C^2E_1^{-1}.E_2$ We have n linear equations with n unknowns $x_1, ..., x_n$.
   - If E_2 is invertible, but E_1 is not invertible, then $C^2 = B^2E_1E_2^{-1}$. We also have n linear equations with n unknowns $x_1, ..., x_n$
   - If both E_1 and E_2 are not invertible but A is invertible, then $A^{-1}E_1 = B^{-1}C, A^{-1}E_2 = C^{-1}B$ We interpret the elements of A^{-1} as the new variables W_i and we end up with $m = 2n$ linear equations in m unknowns. Then we eliminate the new variables to derive n linear equations in the x_i.
   - If none of invertible there occurs a decryption failure. Finally, one computes the plaintext by d. The probability of a decryption failure occurring in the second step is about $1/q$, where q is the cardinality of the underlying field F. It might happen that the linear systems in the second step of the decryption process have multiple solutions $y_1, ..., y_l$. In this case one has to perform the third step for each of these solutions to get a set of possible plaintexts $d_1, ..., d_l$.By encrypting these plaintexts one can test which of them corresponds to the given ciphertext c.

4. Security Analysis

4.1. Rank Attacks: The rank assault can be applied in one of two ways. A demonstration of the first one, known as a MinRank attack or Low Rank attack, The High Rank Attack is the second one. We shall examine these two strategies used to undermine the ABC plan. Without sacrificing generality, let's assume that the public key polynomials and the secret polynomials are both homogeneous quadratic polynomials for the MinRank attack. Let L_1, L_2 be two invertible linear transformations. Let $\overline{Q_1}$.....$\overline{Q_m}$ be the symmetric matrices associated with the public key quadratic polynomials and $Q_1, ..., Q_m$ be the symmetric matrices associate with the secret key quadratic polynomials. Clearly, the rank of Q_i is bounded by $2s$. With the MinRank attack, one tries to find $t_1..., t_m \in k^m$ such that the rank of the linear combinations $\sum_{i=1}^{n} t_i\overline{Q_i}$ is no more than 2s. In order to find such a linear combination, one can choose any vector $v \in k_n$ and try to solve the equations $(\sum_{i=1}^{m} t_i\overline{Q_i})v = 0$ with the

unknown $t_1, ..., t_m$. After finding at least one linear combination of this form, attacker can recover L_2The attacker can recover L_1 and $Q_1, ..., Q_m$ when L_2 is known. More detail about the MinRank attack can be found in [6]. The complexity of this attack against the ABC scheme is $o(q^{\lceil m/n \rceil^{2s}} m^3)$.

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 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.

In our scheme Central map is also form a Cubic Matrix therefore Rank attacks can not be feasible.

4.2. Algebraic Attack: let $\overline{f_1}(x_1 \ldots\ldots\ldots x_n) \ldots\ldots\ldots\ldots\ldots\ldots \overline{f_m}(x_1 \ldots\ldots\ldots\ldots x_n) \in k[x_1 \ldots\ldots\ldots x_n]$ be the public polynomial let $y_1 \ldots\ldots\ldots y_m$ be the ciphertext then try to solve the system of equation

$$\begin{aligned} \overline{f_1}(x_1 \ldots\ldots\ldots x_n) &= y_1 \\ \overline{f_2}(x_1 \ldots\ldots\ldots x_n) &= y_2 \\ \ldots\ldots\ldots\ldots\ldots\ldots & \\ \overline{f_m}(x_1 \ldots\ldots\ldots x_n) &= y_m \end{aligned}$$

In a direct attack (message recovery attack) the attacker tries to solve the public system $F(d) = c$ for the plaintext d. To achieve this, the attacker can use either a Grobner Basis method such as F_4[8] or a system solving algorithm like XL or one of its variants like mutant XL[9, 10, 14, 15]

The adversary faces a system of $m = 2n$ multivariate cubic polynomials in n variables while attempting to undermine our approach. As mentioned in the preceding section, this system was created by multiplying (while ignoring the linear transformations L_1 and L_2) matrices B and C with a matrix A containing linear polynomials selected at random. We assert the following:

Claim: It is simultaneously at least as difficult to solve the cubic public system of our method as it is to solve a multivariate quadratic system with arbitrary cofficients.

To justify this claim, let us assume that an attacker wants to solve the equation $E_2(x) = y$, where $E_2 = AB.C^{-1}$ and y is some matrix in $F^{s \times s}$. Let us further assume that an oracle 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 $\overline{C} \in F^{s \times s}$ with $\overline{C} = C(x)$. 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 $A(x) = y.\overline{C}^{-1}$. But to get the values of $(x_1, ..., x_n)$, the attacker still has to solve a system of multivariate quadratic equations with randomly chosen coefficients.

The following is a far more compelling heuristic argument. Let's identify the matrix's polynomial entries. A by $A_{ij}(x)$ and the polynomial entries of E_1 and E_2 by $E_{1,ijk(x)}$ and $E_{2,ijk(x)}$. And we denote the entries of B and C by $B_{ij(x)}$ and $C_{ij(x)}$ respectively. Clearly we have that

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

It is generally accepted that, in the case of quadratic systems, the structure of the ideal produced by the homogeneous section of highest degree, namely the degree 2 part of the polynomials, truly determines the complexity of the system's solution.

In our situation, this indicates that the system's complexity is

$$A_{ij}(x) = D_{ij}$$

is actually controlled by the structure of the ideal produced by the homogeneous polynomials$\overline{A}_i j(x)$, which are the quadratic component of$A_i j(x)$. This I_A. meets our standards.

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

$$\overline{E}_{1,ij}(x) = \sum_{l=1}^{s} \overline{A}_{il}(x)\ \overline{B}_{lj}^{-1}(x)\overline{C}_{lk}(x)$$
$$\overline{E}_{2,ij}(x) = \sum_{l=1}^{s} \overline{A}_{il}(x)\overline{B}_{lj}(x)\overline{C}_{lk}^{-1}(x)$$

where the homogeneous linear components of $B_i j$ and $C_i j$, respectively, are denoted by $\overline{B}_{ij}$ and $\overline{C}_{ij}$. This is the I_E of the ideal. Now that we have examined the sources of this ideal, we can say with certainty that

$$I_E \subset I_A$$

Furthermore, since $\overline{B}_{ij}(x)$ and $\overline{C}_{ij}(x)$ are only linear functions, the generators of I_E are nothing more than elements in the space spanned by the elements produced in the first phase of the XL method when applied to I_A. From this angle, we therefore hypothesise that the complexity of solving the public systems of our scheme should be generally or precisely asymptotically harder or at least as difficult as solving a quadratic system with randomly chosen coefficients of size $n \times n$ (when s is too small it might be different).

Although this heuristic analysis is highly speculative, it is really intriguing since it suggests that perhaps we might extract a particular type of verifiable We have never seen security for our new system, which is something fresh.

5. Parameter Proposals

We suggest the following settings for our cubic version of the Simple Matrix encryption method based on our security research that was described in the preceding section. For the fields $GF(2^8)$ and $GF(2^{16})$, we recommend being cautions.

1. $s = 7$ for a security level of 80 bit and
2. $s = 8$ for a security level of 100 bit

Therefore, at the phase of solving F_4 when $s = 7$, we can estimate the number of homogeneous monomials of highest degree by

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

Each polynomial's number of non-zero monomials is provided by

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

The complexity of a direct attack against our cubic Simple Matrix Encryption technique can therefore be estimated by

$$Complexity_{directattack}(s = 7) \geq 3.\tau.T^2 \geq 2^{88}.$$

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

$$Complexity_{directattack}(s = 8) \geq 3.\tau.T^2 \geq 2^{102}$$

Table below displays the cubic Simple Matrix encryption scheme's key sizes for our four parameter sets along with the likelihood that the decryption operation will fail.

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

security level(bit)	parameters (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	$2^{-8}4$
	(GF (2^{16}),7,49,98)	784	1,568	4,230	145.4	2^{-16}
100	(GF (2^8),8,64,128)	512	1024	5,988	154	2^{-8}
	(GF (2^{16},8,64,128)	1024	2048	11,976	308	2^{-16}

6. Efficiency of Our Scheme

In this section, we shall examine the effectiveness of our scheme's decryption with that of the HFE challenge 1 by Patarin [4]. The algebraic attack was used to defeat this HFE [12].In this HFE system, J. Patarin selected the following parameters: $q = 2, n = 80$ the degree of central map is 96. Let $P(x)$ be the central map of HFE, the main computation of decryption is to solve the equation $P(x) = y$ over the finite field of 2^{80} elements. In [13], J.Patarin estimated that the complexity of solving this equation is about $O(d^2n^3)$ or $O(dn^3 + d^3n^2)$ depending on the chosen algorithms, where

d is the degree of $P(x)$. Thus the decryption process needs about 6.4×10^9 times field multiplication over the finite field of 2^{80} elements.

For the proposed parameters of the given scheme above, $q = 2^8$, $n = 49$ $m = 98$, and $q = 2^{16}$, $n = 49$ $m = 98$ taking $q = 2^8$, $n = 64$ $m = 128$ and $q = 2^{16}$, $n = 64$ $m = 128$ the steps of decryption were presented in section3. The computation of step 1) and step3) of decryption are very fast. The main computation of decryption is step 2), solving a set of linear equations. Therefore, we only need about $128^3 = 2^{21} \approx 2.110^6$ times field multiplications over the finite field of 2^8 elements for decryption. It is much faster than HFE scheme.

7. Conclusion

In this research, we suggested a new central map for the PQCrypto 2013 [5] cubic Simple Matrix encryption technique. The original Simple Matrix technique is made considerably more secure using matrix A, whose components are randomly selected linear polynomials. Rank attacks against our scheme are totally inconceivable according to our structure. In addition, we suggest that breaking our method using algebraic attacks is at least as challenging as cracking the Cubic Simple Matrix encryption scheme by solving a quadratic system with randomly chosen coefficients. Future work will focus on reducing the probability of decryption errors and expressing our security claim.

Acknowledgements:
The authors would like to thank the anonymous reviewrs for their constructive comments and suggestions

Conflicts of Interest: The authors declare no conflict of interest.

References

[1] J. Ding, A. Petzolt and L. C. Wang, *The Cubic Simple Matrix Encryption scheme PQCrypto*, **8772** (2014), 76-78.

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

[3] T. Matsumoto and H. Imai, *Public quadratic polynomial-tuples for efficient signature verification and message-encryption*, In Gunther, C.G. (ed.) EUROCRYPT, Springer, Heidelberg, **330** (1988), 419–453.

[4] J. Patarin, *Hidden fields equations (HFE) and isomorphisms of polynomials (IP)*, Two new families of asymmetric algorithms, In Maurer, U.M. (ed.) EUROCRYPT, Springer, Heidelberg **1070** (1996) 33–48.

[5] C. Tao, A. Diene, S. Tang and J. Ding, *Simple Matrix scheme for Encryption Post Quantum Cryptography-PQCrypto*,Proceeding, (2014), 231- 242.

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

[7] J. Ding, Gower and J. E. Schmidh, *Multivariate Public Key Cryptosystems* , Springer (2006).

[8] J.C. Faugere, *A new efficient algorithm for computing Grobner bases(F4)*, Journal of Pure and Applied Algebra **139** (1999), 61-88.

[9] N.T. Curtois, A. Klimov, J. Patarin and A. Shamir, *Efficient algorithms for solving overdefined systems of multivariate polynomial equations*, In Preneel, B. (ed.) EUROCRYPT, Springer Heidelberg, **1807** (2000) 392-407.

[10] M.S.E. Mohamed and W.S.A.E. Ding and J. Buchmann and J. MXL2, *Solving polynomial equations over GF(2) using an improved Mutant strategy*, In Buchmann, J., Ding, J. (eds.) PQCrypto, Springer Heidelberg, **5299** (2008), 203-215.

[11] A. Kipnis, A. Shamir, *Cryptanalysis of the HFE public keycryptosystem by relinearization*, In Wiener, M. (ed.) CRYPTO, Springer Heidelberg, **1666** (1999), 19-30.

[12] J. Patarin, *Hidden fields equations (HFE) and isomorphisms of polynomials (IP): Two new families of asymmetric algorithms*, In Maurer, U.M. (ed.) EUROCRYPT, Springer, Heidelberg, **1070** (1996), 33–48.

[13] R. Rivest, A. Shamir and L.M. Adleman, *A method for obtaining digital signatures and public-key cryptosystems*, Communications of the ACM, **21(2)**, 120–126.

[14] J. A. Buchmann, J. Ding and M. S. E. Mohamed, *MutantXL Solving multivariate polynomial equations for cryptanalysis*, Symmetric Cryptography, **09031** (2009).

[15] M.S.E. Mohamed, D. Cabarcas, J. Ding, J. Buchmann, Bulygin and S. MXL3, *An efficient algorithm for computing gr¨obner bases of zero-dimensional ideals*. In Lee, D., Hong, S., et al. (eds.) ICISC, Springer, Heidelberg, **5984** (2010), 87–100.

SUMAN LATA VERMA, Department of Mathematics, Govt. J. Y. Chhattisgarh College Raipur (C.G.), India
e-mail: vermasuman2101@gmail.com

HEMLAL SAHU, Department of Mathematics, Govt. J. Y. Chhattisgarh College Raipur (C.G.), India
e-mail: hemlalsahu@gmail.com