British Journal of Mathematics & Computer Science
5(5): 613-636, 2015, Article no.BJMCS.2015.046
ISSN: 2231-0851

SCIENCEDOMAIN *international*
www.sciencedomain.org

Modified Matrix Modular Cryptosystems

S. K. Rososhek[1*]

[1]Faculty of Mathematics and Mechanics, Tomsk State University, Lenin Street, 36, 634010, Tomsk, Russia.

Article Information

DOI: 10.9734/BJMCS/2015/14321
Editor(s):
(1) Qiankun Song, Department of Mathematics, Chongqing Jiaotong University, China.
Reviewers:
(1) Anonymous, University of Malakand.KPK. Pakistan.
(2) Anonymous, China University of Mining and Technology, China.
Complete Peer review History: http://www.sciencedomain.org/review-history.php?iid=729&id=6&aid=7094

Original Research Article

Received: 26 September 2014
Accepted: 07 November 2014
Published: 06 December 2014

Abstract

The Basic Matrix Modular Cryptosystem (BMMC) is a public-key cryptosystem, which uses some matrix modular exponentiations in the matrix ring over the residue ring modulo n. The aim of this article is to decrease the number of these exponentiations and consequently to accelerate the execution of encryption algorithm. There are two ways to reach this aim. First way is to determine the large abelian subgroup in general linear group over the large residue ring and to choose the session keys in this subgroup, what will be to give the encryption without exponentiations. Other way is to use random integral exponent of the given matrix in the public key as session key and this will be the only exponentiation in encryption algorithm. A discussion about the security of built modifications made in the article shows that the level of security is high enough for an appropriate choice of parameters of the cryptosystems, namely, the lower bound for the selection of secure modulus n is 40-bit integer. Both modified cryptosystems are faster than BMMC and balanced with respect to a pair of security – efficiency, and BMMC is much faster than RSA.

Keywords: Public key cryptosystem, matrix group, residue ring, automorphism.

1 Introduction

Security of some present-day public-key cryptosystems is based on computational complexity of some number-theoretical problems. Two of these problems are used most often: the integer

**Corresponding author: rososhek@list.ru;*

factorization problem and the discrete logarithm problem. These problems ensure the security of the RSA and ElGamal cryptosystems, as well as of the corresponding digital signature schemes [1]. In [2], randomized polynomial-time algorithms for computing discrete logarithms and integer factoring were presented for the quantum computer. Nevertheless, some alternatives should be proposed. One of possible approaches is to replace number-theoretical cryptosystems by such algebraic cryptosystems that would be resistant to an attack on a quantum computer.

Let us now consider some scheme of cryptosystems, namely, cryptosystems of group rings. In the author's work [3], [3a] a scheme of group ring cryptosystems was proposed. The idea to apply group rings in cryptography is based on the fact that if we fix the cardinality of a finite ring R, the cardinality of the group ring RG for a finite group G is an exponent of the cardinality of the group G. Then, a legal user can perform cryptographic transformations separately in the ring R and in the group G using polynomial algorithms and the illegal user has to solve computationally difficult problems in the group ring RG.

In [4] some generalization of group ring cryptosystem is considered in the case of quasigroup ring. In [5] were proposed theoretical attacks on cryptographic schemes using automorphisms in the case of cryptosystems of the matrix rings over finite-dimensional algebras. And although it is questionable possibility of practical realization of these attacks to cryptosystems in the matrix ring over the ring of residues modulo n, this modulus is better to choose as a composite number.

The Basic Matrix Modular Cryptosystem (BMMC) is a public key cryptosystem, which was developed in [6]. In the most important case for the use of public-key cryptosystems, namely, key exchange protocols for symmetric ciphers, such as AES, the key length is usually equal to 128 or 256 bits. Protocol using BMMC was developed for the key exchande in [7].

BMMC realization needs three matrix modular exponentiations for key generation, three exponentiations under encryption and two exponentiations under decryption for every data block. One may to accelerate encryption by decreasing the number of exponentiations. To reach this aim it is necessary to explore the centralizer of random matrix in the general linear group over residue ring. The structure of this centralizer is unknown in general case. There are two ways for the choice of the random element of centralizer. One way is to use a fixed large abelian subgroup G in a such manner that one may easily to choose the random element x in G, then other random element y in G will be an element of centralizer of x. Other way is to compute the random integral exponent y of the fixed element x in the general linear group over residue ring, then y will be in a centralizer of x and that is only exponentiation needs for encryption. Two different modifications of BMMC are based on these two approaches, namely, Modified Matrix Modular Cryptosystem One (MMMC1) and Modified Matrix Modular Cryptosystem Two (MMMC2). Both cryptosystems provide faster encryption than BMMC but their security have to be explored. On the other hand, the mathematical basis of these cryptosystems security is the same hard computational problem unlike BMMC whose security is based on the following two computationally hard mathematical problems [6].

The transformation problem (two-factor conjugacy problem).

Let a matrix P_2 be conjugated with an unknown integral power of a matrix P_1 for two given matrices $P_1, P_2 \in GL_2(\mathbb{Z}_n)$. Find all solutions of the equation with two unknowns Z and y:

$$ZP_2Z^{-1} = P_1^y,$$

where $Z \in GL_2(\mathbb{Z}_n),\ -f(n) < y < f(n)$, y is integer, $f(n)$ be a cardinality of the group $GL_2(\mathbb{Z}_n)$.

The hybrid problem.

Find all solutions of the equation with two unknowns Y and x

$$Y^x = Z_0,$$

where $Y, Z_0 \in GL_2(\mathbb{Z}_n)$, $-f(n) < x < f(n)$, x is integer, $f(n)$ be a cardinality of the group $GL_2(\mathbb{Z}_n)$.

Let us also consider two particular cases of hybrid problem:

a) The discrete logarithm problem in a cyclic subgroup of the group $GL_2(\mathbb{Z}_n)$

Let $H = \langle Y_0 \rangle$ be a fixed cyclic subgroup of order j of the group $GL_2(\mathbb{Z}_n)$ with the generator Y_0 and $M \in H$ be an arbitrary element. Find the unique solution $x = x_0$ of the equation

$$Y_0^x = M,$$

where x is an integer such that $0 \le x < j$.

b) The problem of extracting a root of the ith power in the group $GL_2(\mathbb{Z}_n)$ (the matrix RSA problem)

Let $M \in GL_2(\mathbb{Z}_n)$ be an arbitrary element, i_0 be a fixed integer satisfying the condition $0 \le i_0 < f(n)$ and $GCD(i_0, f(n)) = 1$.

Find all solutions of the equation with a single unknown Y:

$$Y^{i_0} = M,$$

$$Y \in GL_2(\mathbb{Z}_n).$$

According to the problem b), in turn, one can also discern the following problem.

The problem of square-root extraction in $GL_2(\mathbb{Z}_n)$

Find all solutions of the equation with a single unknown *Y*:

$$Y^2 = M,$$

where $Y, M \in GL_2(\mathbb{Z}_n)$.

In the case of MMMC1 and in the case of MMMC2 it is a "random salt" conjugation problem. Classical conjugation problem is the following: for the given elements *A*, *B* in the group *G* to find element *X* in the same group such that

$$X^{-1}AX = B.$$

The "random salt" conjugation problem is the following: for the given matrices *A, B* in the matrix modular ring $M_2(\mathbb{Z}_n)$ over the residue ring $\mathbb{Z}_n$ to find invertible matrix *X* in the same ring and random fixed unit α in the residue ring $\mathbb{Z}_n$ by modulo *n* such that

$$X^{-1}AX = \alpha B.$$

The random fixed "salt" α can be found only under brute force attack and for large enough modulus *n* this problem is becoming intractable.

It should be noted, that some other algebraic cryptosystems are given in [8-13].

The paper is organized as follows: after the Introduction (section 1) is described an abelian subgroup *G* of the group $GL_2(\mathbb{Z}_n)$ (section 2), then briefly is described the BMMC (section 3). In sections 4 and 5 are given the description of MMMC1 and an example of computations, and in sections 6 and 7 are given MMMC2 description and example of computations. The security and efficiency of both modifications of BMMC studied in sections 8 and 9. In section 10 are given the conclusions.

2 Subgroup *G* of the Group $GL_2(\mathbb{Z}_n)$

Let *G* be the following set of 2×2 – matrices:

$$G = \left\{ \begin{pmatrix} a & b \\ b & a \end{pmatrix} \middle| a, b \in \mathbb{Z}_n \text{ and } (a^2 - b^2) \in \mathbb{Z}_n^* \right\},$$

$\mathbb{Z}_n^*$ is an unit group of the residue ring $\mathbb{Z}_n$ modulo *n*.

It is easy to verify that G is an abelian subgroup of the group $GL_2(\mathbb{Z}_n)$:

1) for $K=\begin{pmatrix} a & b \\ b & a \end{pmatrix}\in G$ and $L=\begin{pmatrix} c & d \\ d & c \end{pmatrix}\in G$ we have that

$$M = KL \in G,$$

because

$$KL=\begin{pmatrix} ac+bd & ad+bc \\ ad+bc & ac+bd \end{pmatrix}$$

and determinant of KL is

$$\det(KL) = (\det K)(\det L) \in \mathbb{Z}_n^*;$$

2) for $K=\begin{pmatrix} a & b \\ b & a \end{pmatrix}\in G, \det K = \gamma \in \mathbb{Z}_n^*$ we have that $K^{-1}\in G$, because

$$K^{-1}=\begin{pmatrix} \gamma^{-1}a & -\gamma^{-1}b \\ -\gamma^{-1}b & \gamma^{-1}a \end{pmatrix}, \det K^{-1} = \gamma^{-1} \in \mathbb{Z}_n^*;$$

3) for $K=\begin{pmatrix} a & b \\ b & a \end{pmatrix}\in G$ and $L=\begin{pmatrix} c & d \\ d & c \end{pmatrix}\in G$ we have that $KL = LK$, because

$$KL=\begin{pmatrix} ac+bd & ad+bc \\ ad+bc & ac+bd \end{pmatrix}, LK=\begin{pmatrix} ca+db & cb+da \\ cb+da & ca+db \end{pmatrix}.$$

Let *a, b* be the random elements of the ring $\mathbb{Z}_n$ and let

$$M=\begin{pmatrix} a & b \\ b & a \end{pmatrix}$$

be the corresponding element in the ring $M_2(\mathbb{Z}_n)$. What is the probability of the case that *M* is not in the group *G*? In two special cases of *n* we may to give answer on this question.

Look at these two particular cases of *n:*

1) $n = p^r$, p is a prime number, $2 \le r$ is an integer

Or

2) $n = pq$, p and q are primes.

The cardinality of the residue ring and its unit group in both cases is the following:

1) $\left|\mathbb{Z}_n\right| = p^r, \left|\mathbb{Z}_n^*\right| = \varphi(n) = p^{r-1}(p-1)$

or

2) $\left|\mathbb{Z}_n\right| = pq, \left|\mathbb{Z}_n^*\right| = \varphi(n) = (p-1)(q-1)$,

$\varphi(n)$ is an Euler function.

Then the probability P of the case that matrix M is not in the group G is the following:

1) $1 - \frac{\varphi(n)}{n} = 1 - \frac{p^{r-1}(p-1)}{p^r} = \frac{1}{p}$

or

2) $1 - \frac{\varphi(n)}{n} = 1 - \frac{(p-1)(q-1)}{pq} = \frac{1}{p} + \frac{1}{q} - \frac{1}{pq}$.

If bit length of the primes p and q will be 80 bits or bigger, then we have:

1) $P \le 2^{-80}$

or

2) $P \le 2^{-79}$.

In both cases probability P becomes negligible small and therefore one may to suppose that in these two cases random matrix M over $\mathbb{Z}_n$

$$M = \begin{pmatrix} a & b \\ b & a \end{pmatrix}, a, b \in \mathbb{Z}_n$$

with overwhelming probability is in the group G.

Note. In the case if modulus n is 64-bit integer one needs to verify that random matrix

$$M=\begin{pmatrix} a & b \\ b & a \end{pmatrix}, a,b\in\mathbb{Z}_n$$

belongs to the group *G*.

3 Basic Modular Matrix Cryptosystem (BMMC) [6]

Let us consider the matrices in the matrix ring $M_2(\mathbb{Z})$ and its unit group $Gl_2(\mathbb{Z})$, which contains the free subgroup $G(\alpha,\beta,\gamma)$ of rank 3 with free generators

$$A(\alpha)=\left(\begin{array}{c|c} 1 & 0 \\ \hline \alpha & 1 \end{array}\right), \quad B(\beta)=\left(\begin{array}{c|c} 1 & \beta \\ \hline 0 & 1 \end{array}\right), \quad C(\gamma)=\left(\begin{array}{c|c} 1-\gamma & \gamma \\ \hline -\gamma & \gamma+1 \end{array}\right),$$

where $\alpha,\beta,\gamma\in\mathbb{Z}$ and $|\alpha|\geq 3, |\beta|\geq 3, |\gamma|\geq 3$ [14]. For instance, if $\alpha=\beta=\gamma=3$, we have the matrices

$$A=\left(\begin{array}{c|c} 1 & 0 \\ \hline 3 & 1 \end{array}\right), \quad B=\left(\begin{array}{c|c} 1 & 3 \\ \hline 0 & 1 \end{array}\right), \quad C=\left(\begin{array}{c|c} -2 & 3 \\ \hline -3 & 4 \end{array}\right),$$

which generate the free group $G=G(3,3,3)$ of rank 3.

3.1 Key Generation

Alice doing the following:

1) picks the random large positive integer *n;*
2) picks the random words *W(X)* and *W(U)* in the alphabet $A^{\pm 1}$, $B^{\pm 1}$, $C^{\pm 1}$ in a free rank 3 group with free generators A, B, C;
3) computes the non-commuting matrices X_n and U_n by replacing the symbols A, B, C in the words *W(X),W(U)* with corresponding matrices

$$A=\begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix}, \quad B=\begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}, \quad C=\begin{pmatrix} -2 & 3 \\ -3 & 4 \end{pmatrix}$$

and performing matrix computations by modulo *n* , if X_n and U_n commute, then return to 2) ;

4) let *f(n)* be a cardinality of the group $GL_2(\mathbb{Z}_n)$, then Alice picks random integers *k,s,l* such that $-f(n) \le k,s \le f(n),\ 2 \le l \le f(n), f(n) = |GL_2(\mathbb{Z}_n)|$.
5) Alice public key is $(n, P_1, P_2, P_3) = (n, X, U_n^{-s} X^k U_n^s, U_n^l)$, her private key is (U_n, k, s).

3.2 Encryption

Bob does the following:
for a plaintext $m \in M_2(\mathbb{Z}_n)$ picks $r,t \in \mathbb{Z}_n$ and computes the ciphertext $(C_1, C_2) = (P_3^{-r} P_1^t P_3^r, m \cdot P_3^{-r} P_2^{-t} P_3^r)$.

3.3 Decryption

Alice computes using her private key:
$C_2 U_n^{-s} C_1^k U_n^s = m$.

4 Modified Matrix Modular Cryptosystem One (MMMC1) Descryption

4.1 Key Generation

Alice doing the following:

1) picks a random prime number *p*;
2) makes a choice between
   2.1) picks a random integer $2 \le r$ and computes $n = p^r$;
   2.2) picks a random prime number $q \ne p$ and computes $n = pq$;
3) picks four random integers $a, b, c, d \in \mathbb{Z}_n$;
4) composes two random matrices
$V = \begin{pmatrix} a & b \\ b & a \end{pmatrix}$ and $W = \begin{pmatrix} c & d \\ d & c \end{pmatrix}$;
5) verifies the membership of these matrices to the group *G*;
6) if at least one of these matrices not belongs to *G*, then return to 3).

Note 1. If expected value of modulus *n* is 64-bit integer, then the preferred value of parameter *r* is $r = 2$ and *p* is 32-bit prime number.

Note 2. If modulus *n* is 160-bit integer or bigger, then with overwhelming probability the membership of these matrices to the group *G* is accepted without verification.

7) Alice defines two commuting inner automorphisms of the ring $M_2(\mathbb{Z}_n)$:

$$\alpha : D \to V^{-1}DV,\ \beta : D \to W^{-1}DW$$

for every matrix $D \in M_2(\mathbb{Z}_n).$

Note 3. Matrices $V, W \in G$ and therefore automorphisms α, β commute. Of course this may be checked directly.

8) Alice computes the following automorphisms of the ring $M_2(\mathbb{Z}_n)$:

$$\psi = \alpha^2\beta,\ \varphi = \alpha\beta^2,$$

$$\psi : D \to (V^2W)^{-1}D(V^2W),\ \varphi : D \to (VW^2)^{-1}D(VW^2)$$

for every matrix $D \in M_2(\mathbb{Z}_n).$

Note 4. Automorphisms φ, ψ commute and

$$\psi = \alpha\beta^{-1}\varphi,\ \varphi = \alpha^{-1}\beta\psi.$$

9) Alice picks a random invertible matrix

$$L \in GL_2(\mathbb{Z}_n),$$

such that *L* does not belong to the subgroup *G*;

10) computes matrices:

$$L^{-1},\ \varphi(L),\ \psi(L^{-1});$$

11) Alice public key is

$$(n, \varphi(L), \psi(L^{-1})),$$

private key is

$$(V, W).$$

4.2 Encryption

Bob doing the following:

1) presents the plaintext *m* as a sequence of 2×2- matrices over residue ring $\mathbb{Z}_n$:

$$m^{(1)}, m^{(2)}, \ldots, m^{(N)};$$

2) for every $m^{(i)}, i = 1, 2, \ldots, N$ chooses a random matrix $Y^{(i)} \in G;$

3) defines for every i=1,2,…,N the automorphisms

$$\xi^{(i)} : D \to (Y^{(i)})^{-1} D Y^{(i)}$$

for every $D \in M_2(\mathbb{Z}_n)$;

4) computes for every i=1,2,…,N matrices

$$\xi^{(i)}(\varphi(L)),\ \xi^{(i)}(\psi(L^{-1})),\ m^{(i)}\xi^{(i)}(\varphi(L));$$

5) picks for every $i=1,2,...,N$ random units $\gamma_i \in \mathbb{Z}_n$ ("$salt$") and computes the ciphertext :

$$C = (C^{(1)} \| C^{(2)} \| ... \| C^{(N)}),\ C^{(i)} = (C_1^{(i)}, C_2^{(i)}),$$

$$C_1^{(i)} = \gamma_i^{-1}\xi^{(i)}(\psi(L^{-1})),\ C_2^{(i)} = \gamma_i m^{(i)}\xi^{(i)}(\varphi(L)), i = 1,2,...,N.$$

4.3 Decryption

Alice doing the following:

1) computes for every $i=1,2,...,N$ using her private key:

$$z^{(i)} = \alpha^{-1}\beta(C_1^{(i)}) = \alpha^{-1}\beta(\gamma_i^{-1}\xi^{(i)}(\psi(L^{-1}))),$$

2) computes for every $i=1,2,...,N$ matrices:

$$C_2^{(i)} z^{(i)} = (\gamma_i m^{(i)}\xi^{(i)}(\varphi(L)))z^{(i)} = m^{(i)},$$

3) restores the plaintext m from the matrix sequence $m^{(1)}, m^{(2)}, ..., m^{(N)}$.

Note 5. Decryption correctness proof will be given later for two modifications.

5 Example 1 (MMMC1)

5.1 Key Generation

Alice doing the following:

1) picks the primes p = 5, q = 7 and computes $n = pq = 35$;

2) chooses four random integers in the residue ring $\mathbb{Z}_{35}$:

7, 4, 6, 2;

3) composes the random matrices

$$V = \begin{pmatrix} 7 & 4 \\ 4 & 7 \end{pmatrix},\ W = \begin{pmatrix} 6 & 2 \\ 2 & 6 \end{pmatrix};$$

4) computes $\det V = 33,\ \det W = 32$ and then computes

$$(\det V)^{-1} = 17,\ (\det W)^{-1} = 23,$$

therefore V and W are units in the matrix ring $M_2(\mathbb{Z}_{35})$;

5) defines two automorphisms of the ring $M_2(\mathbb{Z}_{35})$:

$$\alpha : D \to V^{-1}DV,\ \beta : D \to W^{-1}DW$$

for every matrix $D \in M_2(\mathbb{Z}_{35})$;

6) computes the following automorphisms:

$$\psi = \alpha^2\beta,\ \varphi = \alpha\beta^2;$$

7) chooses the random matrix $L \in GL_2(\mathbb{Z}_{35})$:

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

and computes matrix

$$L^{-1} = \begin{pmatrix} 30 & 2 \\ 3 & 34 \end{pmatrix};$$

8) computes matrices:

$$\varphi(L) = (VW^2)^{-1}L(VW^2) = \begin{pmatrix} 34 & 34 \\ 6 & 7 \end{pmatrix},$$

$$\psi(L^{-1}) = (V^2W)^{-1}L^{-1}(V^2W) = \begin{pmatrix} 23 & 24 \\ 16 & 6 \end{pmatrix}.$$

9) Alice public key is

$$\left(n = 35, \varphi(L) = \begin{pmatrix} 34 & 34 \\ 6 & 7 \end{pmatrix}, \psi\left(L^{-1}\right) = \begin{pmatrix} 23 & 24 \\ 16 & 6 \end{pmatrix} \right),$$

private key is

$$\left(V = \begin{pmatrix} 7 & 4 \\ 4 & 7 \end{pmatrix},\ W = \begin{pmatrix} 6 & 2 \\ 2 & 6 \end{pmatrix} \right).$$

5.2 Encryption

Bob doing the following:

1) presents the plaintext as a matrix $m \in M_2(\mathbb{Z}_{35})$:

$m = \begin{pmatrix} 11 & 2 \\ 9 & 3 \end{pmatrix}$;

2) picks the random matrix $Y = \begin{pmatrix} 3 & 5 \\ 5 & 3 \end{pmatrix} \in G$ and computes $Y^{-1} = \begin{pmatrix} 2 & 20 \\ 20 & 2 \end{pmatrix}$;

3) defines automorphism ξ of the ring $M_2(\mathbb{Z}_{35})$:

$\xi : D \to Y^{-1}DY$

for every matrix $D \in M_2(\mathbb{Z}_{35})$;

4) computes matrices:

$\xi(\varphi(L)) = Y^{-1}\varphi(L)Y = \begin{pmatrix} 29 & 24 \\ 16 & 12 \end{pmatrix}$,

$\xi(\psi(L^{-1})) = Y^{-1}\psi(L^{-1})Y = \begin{pmatrix} 13 & 24 \\ 16 & 16 \end{pmatrix}$;

5) picks random unit $\gamma \in \mathbb{Z}_{35} : \gamma = 9,\ \gamma^{-1} = 4$;

6) computes the ciphertext $C = (C_1, C_2)$:

$C_1 = \gamma^{-1}\xi(\psi(L^{-1})) = \begin{pmatrix} 17 & 26 \\ 29 & 29 \end{pmatrix}$, $C_2 = \gamma m \xi(\varphi(L)) = \begin{pmatrix} 9 & 2 \\ 16 & 28 \end{pmatrix}$.

5.3 Decryption

Alice doing the following:

1) computes matrix *z*, using her private key:

$z = \alpha^{-1}\beta(C_1) = \begin{pmatrix} 22 & 26 \\ 29 & 24 \end{pmatrix}$;

2) computes then

$C_2 z = \begin{pmatrix} 11 & 2 \\ 9 & 3 \end{pmatrix} = m.$

6 Modified Matrix Modular Cryptosystem Two (MMMC2) Description

6.1 Key Generation

Alice doing the following:

1) picks a random prime number *p*;

2) chooses one of the following two cases:

2.1) picks a random integer $2 \le r$ and computes $n = p^r$;

2.2) picks a random prime number $q \ne p$ and computes $n = pq$;

3) picks a random matrix $W \in GL_2(\mathbb{Z}_n)$;

4) computes matrices

$$F = W^2,\ H = W^3$$

and

$$F^2H,\ FH^2;$$

5) picks a random matrix $L \in GL_2(\mathbb{Z}_n)$;

6) defines the automorphisms:

$$\alpha : D \to F^{-1}DF,\ \beta : D \to H^{-1}DH$$

and then computes the automorphisms:

$$\psi = \alpha^2\beta,\ \varphi = \alpha\beta^2,$$

for every matrix $D \in M_2(\mathbb{Z}_n)$.

Note 1. Automorphisms $\alpha, \beta, \varphi, \psi$ commute each with other because the corresponding matrices F, H, F^2H, FH^2 are some integral exponents of matrix *W*.

7) Alice computes matrices:

$$FH,\ \varphi(L),\ \psi(L^{-1});$$

8) Alice public key is

$$(n, \varphi(L), \psi(L^{-1}), FH),$$

private key is

$$(F, H).$$

Note 2. The following relations can be easily verified: $\varphi = \alpha\beta^{-1}\psi,\ \psi = \alpha^{-1}\beta\varphi.$

6.2 Encryption

Bob doing the following:

1) presents the plaintext as a sequence of 2×2 - matrices over the residue ring $\mathbb{Z}_n$:

$m^{(1)},...,m^{(N)}$;

2) for every matrix $m^{(i)}, i=1,...,N$ picks random integer k_i and computes matrix

$Y^{(i)}=(FH)^{k_i}$;

3) defines automorphisms:

$\xi^{(i)}: D \to (Y^{(i)})^{-1} D Y^{(i)}$

for every matrix $D \in M_2(\mathbb{Z}_n),\ i=1,...,N$;

4) computes matrices for every *i=1,...,N*:

$\xi^{(i)}(\varphi(L)),\ \xi^{(i)}(\psi(L^{-1})),\ m^{(i)}\xi^{(i)}(\varphi(L))$;

5) picks random units $\gamma_i \in \mathbb{Z}_n^*,\ i=1,...,N$ and computes ciphertext:

$$C = C^{(1)} \| ... \| C^{(N)},$$

$$C^{(i)} = (C_1^{(i)}, C_2^{(i)}), i=1,...,N,$$

$$C_1^{(i)} = \gamma_i^{-1} \xi^{(i)}(\psi(L^{-1})),\ C_2^{(i)} = \gamma_i m^{(i)} \xi^{(i)}(\varphi(L)).$$

6.3 Decryption

Alice doing the following:

1) using her private key computes for every *i=1,...,N*:

$$z^{(i)} = \alpha^{-1}\beta\left(C_1^{(i)}\right) = \alpha^{-1}\beta\left(\gamma_i^{-1}\xi^{(i)}\left(\psi\left(L^{-1}\right)\right)\right);$$

2) computes matrices for every *i=1,...,N*:

$C_2^{(i)} z^{(i)} = m^{(i)}$;

3) restores the plaintext *m* from the matrix sequence

$m^{(1)},...,m^{(N)}$.

Note 3. Automorphisms $\xi^{(i)},\ i=1,...,N$ commute with automorphisms $\alpha, \beta, \varphi, \psi$.

Theorem. The decryption in cryptosystems MMMC1 and MMMC2 is correct.

Proof. Automorphisms $\xi^{(i)},\ i=1,...,N$ commute with automorphisms α and β in both cryptosystems, besides of definitions of one-named automorphisms are quite different in them. Therefore proof deals with both cryptosystems.

Under computations we have the following:

$$\left(\gamma_i m^{(i)}\xi^{(i)}\left(\varphi\left(L\right)\right)\right)z^{(i)} = \left(\gamma_i m^{(i)}\xi^{(i)}\left(\varphi(L)\right)\right)\left(\alpha^{-1}\beta\left(\gamma_i^{-1}\xi^{(i)}\left(\psi\left(L^{-1}\right)\right)\right)\right) =$$

$$= \left(\gamma_i\gamma_i^{-1}m^{(i)}\xi^{(i)}\left(\varphi(L)\right)\right)\left(\xi^{(i)}\left(\alpha^{-1}\beta\left(\psi\left(L^{-1}\right)\right)\right)\right) =$$

$$= m^{(i)}\left(\xi^{(i)}\left(\varphi\left(L\right)\right)\right)\left(\xi^{(i)}\left(\varphi\left(L^{-1}\right)\right)\right) = m^{(i)}\left(\xi^{(i)}\left(\varphi\left(L\right)\varphi\left(L^{-1}\right)\right)\right) =$$

$$= m^{(i)}\xi^{(i)}\varphi\left(LL^{-1}\right) = m^{(i)}\xi^{(i)}\varphi\left(I\right) = m^{(i)}I = m^{(i)}.$$

7 Example 2 (MMMC2)

7.1 Key Generation

Alice doing the following:

1) picks prime number $p = 5$ and computes $n = p^2 = 25$;

2) picks the random matrix $W = \begin{pmatrix} 7 & 3 \\ 5 & 2 \end{pmatrix} \in GL_2(\mathbb{Z}_{25})$;

3) computes matrices $F = W^2 = \begin{pmatrix} 14 & 2 \\ 20 & 19 \end{pmatrix}$, $H = W^3 = \begin{pmatrix} 8 & 21 \\ 10 & 23 \end{pmatrix}$

and also matrices $F^2H = \begin{pmatrix} 23 & 24 \\ 15 & 8 \end{pmatrix}$, $FH^2 = \begin{pmatrix} 6 & 17 \\ 20 & 11 \end{pmatrix}$;

4) chooses the random matrix $L = \begin{pmatrix} 9 & 4 \\ 7 & 3 \end{pmatrix} \in GL_2(\mathbb{Z}_{25})$;

5) defines automorphisms $\alpha, \beta, \varphi, \psi$;

6) computes matrices

$$FH = \begin{pmatrix} 7 & 15 \\ 0 & 7 \end{pmatrix},\ \varphi(L) = \begin{pmatrix} 8 & 24 \\ 17 & 4 \end{pmatrix},\ \psi(L^{-1}) = \begin{pmatrix} 3 & 8 \\ 7 & 10 \end{pmatrix};$$

7) Alice public key is:

$$\left(n=25,\varphi(L)=\begin{pmatrix}8 & 24\\ 17 & 4\end{pmatrix},\psi(L^{-1})=\begin{pmatrix}3 & 8\\ 7 & 10\end{pmatrix},FH=\begin{pmatrix}7 & 15\\ 0 & 7\end{pmatrix}\right),$$

her private key is $\left(F=\begin{pmatrix}14 & 2\\ 20 & 19\end{pmatrix},H=\begin{pmatrix}8 & 21\\ 10 & 23\end{pmatrix}\right).$

7.2 Encryption

Bob doing the following:

1) presents the plaintext as matrix $m=\begin{pmatrix}9 & 16\\ 10 & 5\end{pmatrix}\in M_2(\mathbb{Z}_{25});$

2) picks random *k,* for example, $k=3$ and computes matrix $Y=(FH)^3=\begin{pmatrix}18 & 5\\ 0 & 18\end{pmatrix};$

3) defines automorphism $\xi: D\to Y^{-1}DY$ for every matrix $D\in M_2(\mathbb{Z}_{25});$

4) computes matrices

$$\xi(\varphi(L))=\begin{pmatrix}13 & 14\\ 17 & 24\end{pmatrix},\ \xi(\psi(L^{-1}))=\begin{pmatrix}8 & 13\\ 7 & 5\end{pmatrix},\quad m\xi(\varphi(L))=\begin{pmatrix}8 & 10\\ 15 & 10\end{pmatrix};$$

5) picks $\gamma=7$, then computes

$$C_1=\gamma^{-1}\xi(\psi(L^{-1}))=\begin{pmatrix}19 & 9\\ 1 & 15\end{pmatrix},\ C_2=\gamma m\xi(\varphi(L))=\begin{pmatrix}23 & 20\\ 5 & 20\end{pmatrix},$$

$$C=(C_1,C_2).$$

7.3 Decryption

Alice doing the following:

1) computes matrix *z* using her private key:

$$z=\alpha^{-1}\beta(C_1)=\begin{pmatrix}18 & 2\\ 6 & 16\end{pmatrix};$$

2) computes

$$C_2z=\begin{pmatrix}9 & 16\\ 10 & 5\end{pmatrix}=m.$$

8 Security of MMMC1 and MMMC2

Look at some of the attacks on the cryptosystems MMMC1 and MMMC2.

8.1 A ciphertext only attack

Let $C=(C_1,C_2)$ be a ciphertext for plaintext *m*, then

$C_1=\gamma^{-1}\xi\left(\psi(L^{-1})\right),\ C_2=\gamma m\xi\left(\varphi(L)\right)$

and therefore we come to the equation system with unknowns matrices *m, Y* and unknown unit $\gamma\in\mathbb{Z}_n^*$:

$$\begin{cases} C_1=\gamma^{-1}Y^{-1}\psi(L^{-1})Y, \\ C_2=\gamma mY^{-1}\varphi(L)Y. \end{cases} \qquad (*)$$

For random unit γ cryptanalyst has not another way to solve this equation system as to suppose concrete value $\gamma=\gamma_0$ and to solve the conjugation problem: to find unknown matrix *Y* from the equation:

$$\gamma_0 C_1=Y^{-1}\psi(L^{-1})Y.$$

Rewriting this matrix equation as system of four linear equations with four unknowns cryptanalyst finds the set of solutions, depending on one or more parameters, each of which runs $\mathbb{Z}_n$. Then he inserts each solution $Y=Y_0$ in the second equation of system (*) :

$$C_2=\gamma_0 mY_0^{-1}\varphi(L)Y_0$$

and finds corresponding solution $m=m_0$. Thus, for each fixed γ_0 cryptanalyst receives at least *n* pairs of the form (Y_0,m_0). Because γ_0 accepts $\varphi(n)$ values, the cryptanalyst gets $n\varphi(n)$ triples of the form (γ_0,Y_0,m_0). Consequently, if *n* is not less than 64-bit integer, then check, which of these non less approximately 2^{125} triplets is a true solution, becomes infeasible. Therefore, the lower bound for the selection secure modulus *n* is 40-bit integer, because in that case one needs to check approximately 2^{78} triplets.

8.2 A Known-plaintext Attack

Let $(m^{(1)}, C^{(1)}), \ldots, (m^{(k)}, C^{(k)})$ be the pairs of the form plaintext-ciphertext. Cryptanalyst needs to find unknown plaintext $m^{(k+1)}$ from the corresponding ciphertext $C^{(k+1)}$. In our case for the cryptosystems MMMC1 and MMMC2 encryption uses the new random one-time key *Y* for the new plaintext. Therefore knowledge of previous pairs of the form plaintext-ciphertext gives no information to find the unknown plaintext from the corresponding ciphertext for a new pair.

8.3 A Chosen-plaintext Attack

There are the same arguments as for a known-plaintext attack.

8.4 An Adaptive Chosen-plaintext Attack

There are the same arguments as before.

8.5 A Chosen-ciphertext Attack

Let m^* be a random matrix in the group $GL_2(\mathbb{Z}_n)$, $C = (C_1, C_2)$ be a ciphertext for unknown plaintext *m*. Cryptanalyst Connor computes $m^* C_2$ and offers Alice to decrypt the ciphertext $C^* = (C_1, m^* C_2)$. Then Alice finds corresponding plaintext $m^* m$ and sends it to Connor. Finally Connor computes the initial plaintext as the following: $(m^*)^{-1}(m^* m) = m.$

8.6 Protection from a Chosen Ciphertext Attack

To prevent this attack one has to replace one-sided ciphertext with two-sided ciphertext. Namely, one-sided ciphertext:

$$C = (C_1, C_2),\ C_1 = \gamma^{-1} Y^{-1} \psi(L^{-1}) Y,\ C_2 = \gamma m Y^{-1} \varphi(L) Y$$

is replaced with two-sided ciphertext

$$C = (C_1, C_2),\ C_1 = \gamma^{-1} Y^{-1} \psi(L^{-1}) Y,\ C_2 = \gamma^2 Y^{-1} \varphi(L) Y m Y^{-1} \varphi(L) Y.$$

In this case decryption becames the following:

a) Alice computes $z = \alpha\beta^{-1}(C_1);$

b) then computes $zC_2 z = m.$

The chosen ciphertext attack in this case will not be successful, since the matrices *Y* and *m* in general do not commute. These variants of the MMMC1 and MMMC2 will be called closed cryptosystem variants.

Note. An attack with a chosen ciphertext breaks cryptosystems RSA, ElGamal and Rabin, but attempts to build their modifications resistant to this attack, still resulted in a very inefficient cryptosystems. As we can see, for the matrix modular cryptosystems situation is different, since the closed variant differs from the usual only a few number of matrix multiplications.

9 Comparing Efficiency of Cryptosystems

For comparing the bit complexity of encryption and decryption algorithms considered in the paper cryptosystems start with the known estimates of the bit complexity of basic operations in the residue ring ([1], p. 72, Table 2.5).

Table 2.5. Bit complexity of basic operations in $\mathbb{Z}_n$

Operations		Bit complexity
Modular addition	$(a+b) \bmod n$	$O(\lg n)$
Modular subtraction	$(a-b) \bmod n$	$O(\lg n)$
Modular multiplication	$(ab) \bmod n$	$O((\lg n)^2)$
Modular inversion	$a^{-1} \bmod n$	$O((\lg n)^2)$
Modular exponentiation	$a^k \bmod n, k < n$	$O((\lg n)^3)$

We now find the bit complexity of modular matrix operations used in compared cryptosystems.

9.1 Matrix Modular Multiplication

This operation consists of 8 modular multiplications and 4 modular additions. Considering only the multiplication, we obtain the following estimate of the bit complexity of the matrix modular multiplication: $8(\lg n)^2 C$ -bit operations for some constant C.

For 64-bit *n* we obtain the following estimate:

$8(64)^2 = 2^{15} = 32 \times 2^{10} \approx 3{,}2 \times 10^4 C$ -bit operations for some constant C.

9.2 Matrix Modular Multiplication in the Group *G*

This operation consists of 4 modular multiplications and 2 modular additions. Considering only the multiplication, we obtain the following estimate of the bit complexity of the matrix modular multiplication: $4(\lg n)^2 C$ -bit operations for some constant C.

For 64-bit n we obtain the following estimate:

$4(64)^2 = 2^{14} = 16 \times 2^{10} \approx 1,6 \times 10^4 C$ -bit operations for some constant C.

9.3 Matrix Modular Multiplication by a Scalar

This operation consists of 4 matrix modular multiplications, then we obtain the following estimate of the bit complexity of the matrix modular multiplication: $4(\lg n)^2 C$ -bit operations.
For 64-bit n we obtain the following estimate:

$4(64)^2 = 4 \times 2^{12} = 16 \times 2^{10} \approx 1,6 \times 10^4 C$ -bit operations for some constant C.

9.4 Matrix Modular Inversion

This operation consists of 2 modular multiplications and 1 modular subtraction (computation of the determinant), modular inversion and matrix modular multiplication by a scalar, then we obtain the following estimate of the bit complexity of the matrix modular inversion:

$3(\lg n)^2 + 4(\lg n)^2 = 7(\lg n)^2 C$ -bit operations (modular subtraction is ignored).
For 64-bit n we obtain the following estimate:

$7(64)^2 = 7 \times 2^{12} = 28 \times 2^{10} \approx 2,8 \times 10^4 C$ -bit operations for some constant C.

9.5 Matrix Modular Inversion in the Group *G*

Recall that in group G matrices are of the form $D = \begin{pmatrix} a & b \\ b & a \end{pmatrix}$, then we obtain:

$3(\lg n)^2 + 2(\lg n)^2 = 5(\lg n)^2 C$ -bit operations.

For 64-bit n we obtain the following estimate:
$5(64)^2 = 5 \times 2^{12} = 20 \times 2^{10} \approx 2 \times 10^4 C$ -bit operations for some constant C.

9.6 Matrix Modular Exponentiation

This operation consists of lg n matrix modular multiplications, then we obtain the following estimate of the bit complexity of the matrix modular exponentiation:

$\lg n \cdot 8(\lg n)^2 = 8(\lg n)^3 C$ -bit operations.

For 64-bit n we obtain the following estimate:

$8(64)^3 = 8 \times 2^{18} = 2 \times 2^{20} = 2 \times (2^{10})^2 \approx 2 \times 10^6 C$ -bit operations for some constant C.

Now we can compare the bit complexity of the encryption and decryption discussed in the paper cryptosystems.

We first consider as a reference point the cryptosystem RSA. Encryption and decryption in RSA consist of only modular exponentiation, then their bit complexity are $(\lg n)^3 C$ -bit operations, for 1024-bit n and 1024-bit plaintext block we obtain the following estimate:

$1024^3 = (2^{10})^3 \approx (10^3)^3 = 10^9 C$ -bit operations for some constant C.

Note. There are more efficient modifications of the RSA but we are now considering only classical cryptosystem.

As another reference point consider fast ciphers, namely, symmetric or stream ciphers. It is known that these ciphers are much faster than public-key cryptosystems, according to some estimates, about 1000 times faster. So there is a very roughly estimate of the bit complexity of fast ciphers as a range of $10^4 - 10^6$ -bit operations. Now let's see whether the fall in this range at least some of the discussed in the paper cryptosystems.

1. BMMC

Encryption in BMMC consists of 3 matrix modular exponentiations (can be neglected other operations), then the bit complexity of the BMMC encryption is the following:

$3 \cdot 8(\lg n)^3 = 24(\lg n)^3 C$ -bit operations for some constant C.

For 64-bit n and 256-bit plaintext block we obtain the following estimate:

$24(64)^3 = 24 \cdot 2^{18} = 6 \cdot 2^{20} \approx 6 \times (10^3)^2 = 6 \times 10^6 C$ -bit operations.

Decryption in BMMC consists of 2 matrix modular exponentiations, then the bit complexity of the BMMC decryption is the following:

$2 \cdot 8(\lg n)^3 = 16(\lg n)^3 C$ -bit operations.

For 64-bit n and 512-bit ciphertext block we obtain the following estimate:

$16(64)^3 = 16 \cdot 2^{18} = 4 \times 2^{20} \approx 4 \times 10^6 C$ -bit operations.

2. MMMC1

Encryption in MMMC1 consists of 5 matrix modular multiplications, 1 matrix modular inversion in the group G, 2 matrix modular multiplications by scalar and 1 modular inversion. Then the bit complexity of the MMMC1 encryption is the following:

$5 \cdot 8(\lg n)^2 + 5(\lg n)^2 + 2 \cdot 4(\lg n)^2 + (\lg n)^2 = 54(\lg n)^2 C$ -bit operations.

For 64-bit *n* and 256-bit plaintext block we obtain the following estimate:

$54(64)^2 = 54 \cdot 2^{12} = 216 \times 2^{10} \approx 216 \times 10^3 \approx 2,2 \times 10^5 C$ -bit operations.

Decryption in MMMC1 consists of 3 matrix modular multiplications, 1 matrix modular multiplication in the group *G* and 2 matrix modular inversions in the group *G*. Then the bit complexity of the MMMC1 decryption is the following:

$3 \cdot 8(\lg n)^2 + 4(\lg n)^2 + 2 \cdot 5(\lg n)^2 = 38(\lg n)^2 C$ -bit operations.

For 64-bit *n* and 512-bit ciphertext block we obtain the following estimate:

$38(64)^2 = 38 \cdot 2^{12} = 142 \times 2^{10} \approx 142 \times 10^3 \approx 1,4 \times 10^5 C$ -bit operations.

3. MMMC2

Encryption in MMMC2 consists of only matrix modular exponentiation (can be neglected other operations), then the bit complexity of the MMMC2 encryption is the following:

$8(\lg n)^3 C$ -bit operations for some constant C.

For 64-bit *n* and 256-bit plaintext block we obtain the following estimate:

$8(64)^3 = 8 \cdot 2^{18} = 2 \cdot 2^{20} \approx 2 \times 10^6 C$ -bit operations.

Decryption in MMMC2 consists of 4 matrix modular multiplications and 2 matrix modular inversions. Then the bit complexity of the MMMC2 decryption is the following:

$4 \cdot 8(\lg n)^2 + 2 \cdot 7(\lg n)^2 = 46(\lg n)^2 C$ -bit operations.

For 64-bit *n* and 512-bit ciphertext block we obtain the following estimate:

$46(64)^2 = 46 \cdot 2^{12} = 184 \cdot 2^{10} \approx 184 \times 10^3 \approx 1,8 \times 10^5 C$ -bit operations.

So, BMMC is a bit out of fast ciphers range, although significantly faster RSA. MMMC2 encryption is faster BMMC but a little outside of fast ciphers range and MMMC2 decryption falls within the range of fast ciphers. Finally, encryption and decryption in MMMC1 fall within a range of fast ciphers.

10 Conclusion

In paper two modifications of Basic Matrix Modular Cryptosystem (BMMC) are developed. Both cryptosystems are faster than the BMMC and balanced with respect to a pair of security-efficiency.

Replacing BMMC in non-commutative analogue of the Diffie-Hellman key exchange protocol on the one of the modified matrix cryptosystems, we obtain plaintext (symmetric cipher key) as a 2×2 matrix over the residue ring $\mathbb{Z}_n$ with 64-bit modulus *n* for 256-bit key, and for the case of the 128-bit key may be used advance arrangements half of the 256-bit string. As shown in the paper, for such modulus *n* modified matrix modular cryptosystems are fast and secure, they can be used in various applications, especially given the fact that the lower bound for the security of the

modified matrix modular cryptosystems is a 40-bit modulus n of the residue ring $\mathbb{Z}_n$. The fastest of the three ciphers discussed in the paper is a cryptosystem MMMC1, it is near the speed of encryption and decryption to symmetric and stream ciphers. Decryption in MMMC2 too fast, but encryption is slightly inferior in speed to MMMC1. Finally, the encryption and decryption in the BMMC inferior in speed to MMMC2, although much faster RSA.

Competing Interests

Author has declared that no competing interests exist.

References

[1] Menezes A, van Ooshot P, Vanstone S. Handbook of applied cryptography, Toronto: CRC Press; 1996.

[2] Shor PW. Algorithms for quantum computation: discrete logarithm and factoring. Proceedings of the IEEE 35th Communications Annual Symposium on Foundations of Computer Science, Santa Fe, NM: Springer-Verlag. 1994;124-134.

[3] Rososhek SK. Cryptosystems in automorphism groups of group rings of Abelian groups. Fundamentalnaya I prikladnaya matematica. 2007;13:157-164. (Russian).

[3a] Rososhek SK. Cryptosystems in Automorphism groups of group rings of Abelian groups. Journal of Mathematical Sciences. 2008;154:386-391.

[4] Gribov AN, Zolotykh PA, Mikhalev AV. A construction of algebraic cryptosystem over the quasigroup ring. Mathematical Aspects of Cryptography. 2010;1:23-32. (Russian).

[5] Romankov VA. Cryptanalysis of some cipher schemes used the automorphisms. Applied Discrete Math. 2013;3(21):35-51. (Russian).

[6] Rososhek SK. New practical algebraic public-key cryptosystem and some related algebraic and computational aspects. Applied Mathematics. 2013;4(7):1043–1049.

[7] Rososhek SK, Gorbunov ES. Non-commutative analogue of Diffie-Hellman protocol in matrix ring over the residue ring. International Journal of Computers and Technology. 2013;11(10):3051-3059.

[8] Paeng SH, Ha KC, Kim JH, Chee S, Park C. New public-key cryptosystem using finite non-abelian groups, Proc. Crypto 2001, Lect. Notes in Comp. Science. 2001;2139:470-485.

[9] Mahalanobis A. A simple generalization of the ElGamal cryptosystem to non-abelian groups. Communic. in Algebra. 2008;36:3878-3889.

[10] Dehornoy P. Braid group cryptography, Contemp. Math. 2004;360:5-33.

[11] Myasnikov A, Shpilrain V, Ushakov A. Group-based cryptography, Basel-Berlin-New York: Birkhouser Verlag. 2008;183.

[12] Myasnikov A, Shpilrain V, Ushakov A. Non-commutative cryptography and complexity of group-theoretic problems, Amer. Math. Soc. Surveys and Monographs, Providence, RI: Amer. Math. Soc. 2011;385.

[13] Blackburn SR, Cid C, Mullan C. Cryptanalysis of three matrix-based key establishment protocols. Journal of Math. Cryptography. 2011;5:159-168.

[14] Merzlyakov YI. Matrix representations of free groups. Doklady Akademii Nauk. 1978;238:527-533. (Russian).

Peer-review history:
The peer review history for this paper can be accessed here (Please copy paste the total link in your browser address bar)
www.sciencedomain.org/review-history.php?iid=729&id=6&aid=7094