ISSN 2277-3061

Noncommutative analogue of Diffie-Hellman protocol in matrix ring over the residue ring

S.K. Rososhek, E.S. Gorbunov
Associate Professor of Faculty of Mathematics and Mechanics, Tomsk State University,
Lenin av., 36, Tomsk, Russia
rososhek@list.ru
Faculty of Mathematics and Mechanics, Tomsk State University
Lenin av., 36, Tomsk, Russia
ghostman23@mail.ru

ABSTRACT

Classical Diffie-Hellman protocol of the key establishment was the basis of the development of several key exchange protocols. But this protocol is not secure and it is not protected against the “man in the middle” attack. The purpose of this article is to offer a secure and practical noncommutative analogue of the Diffie–Hellman protocol that is reliably protected not only against “man in the middle” attack but also against the quantum computer attack.

Keywords: public key cryptosystem, key exchange protocol, attacks on the protocol.

Council for Innovative Research

Peer Review Research Publishing System

Journal: INTERNATIONAL JOURNAL OF COMPUTERS & TECHNOLOGY

Vol 11, No.10

editor@cirworld.com

www.cirworld.com, member.cirworld.com

1. INTRODUCTION

Security of the most popular present-day public key cryptosystems is based on the computational complexity of the some problems in number theory. Two of these problems are the most common: the factorization problem and the discrete logarithm problem. Nevertheless, the creation of the sufficiently powerful quantum computer will make these cryptosystems useless, since at present there are quantum algorithms that solve these problems in polynomial time [1]. Due to the need to develop new approaches in cryptography to protect against the potential threat posed by the quantum computer works have appeared that use noncommutative algebraic objects (groups and rings) as a platform for building cryptosystems. This line of research has been called – noncommutative algebraic cryptography. At present, the most active area of research is a combination of combinatorial group theory and linear groups to construct cryptosystems, in fact, proposed a general scheme for such cryptosystems [2], [3], [4].

Developed a general scheme of noncommutative analogue of Diffie–Hellman key establishment protocol in a non-Abelian group [5], but this scheme uses an unrealistic assumption of the existence in the group two element-wise commuting subgroups, which are the secrets of two users, so the development of the noncommutative analogue of the Diffie–Hellman key establishment protocol, which has perspective of practical use, there is an urgent problem of noncommutative algebraic cryptography. In [6]–[13] have been proposed public key cryptosystems, using as a platform for building cryptosystems noncommutative groups and rings. However, in [14], [15] discovered vulnerabilities in some of these cryptosystems, the presence of which indicates the need for further research in this area. Therefore there is a need for secure noncommutative public key cryptosystems, which could be used in the construction of noncommutative analogue of the Diffie–Hellman key establishment protocol for protection against "man in the middle" attack. Just such a cryptosystem is BMMC (Basic Matrix Modular Cryptosystem) [16].

2. BMMC (Basic Matrix Modular Cryptosystem)

I) Key generation

User Alice performs the following actions.

1. Selects large positive integer *n*.

2. Chooses the random words *W* (*X*) and *W* (*U*) in the alphabet of characters $A^{\pm 1}, B^{\pm 1}, C^{\pm 1}$, where *A*, *B*, *C* are the free generators in a free non-Abelian group of rank 3.

3. Computes the noncommuting matrices X_n, U_n according to *W* (*X*) and *W* (*U*), respectively, where the characters A, B, C are replaced in these words on the matrices $A = \begin{pmatrix} 1 & 0 \\ 3 & 1 \end{pmatrix}, B = \begin{pmatrix} 1 & 3 \\ 0 & 1 \end{pmatrix}, C = \begin{pmatrix} -2 & 3 \\ -3 & 4 \end{pmatrix}$ respectively.

All computations are done modulo *n*.

If X_n, U_n commute, then return to step 2.

4. Chooses random integers *k, s,* l that satisfy

$$-f(n) + 2 \le k, s \le f(n) - 2, 2 \le l \le f(n) - 2,$$

$f(n) = |GL_2(\mathbf{Z}_n)|$ is the order of the general linear group over the ring $\mathbf{Z}_n$ of residues modulo *n*.

Alice obtains a pair of keys P_A, S_A ,

public key $P_A = \; n, P_1, P_2, P_3$, $P_1 = X_n$, $P_2 = U_n^{-s} X_n^k U_n^s$, $P_3 = U_n^l$,

private key $S_A = (U_n, s, k)$.

Note

The order of the general linear group $GL_2(\mathbf{Z}_n)$ in particular cases can be computed by the formulas [16]:

if $n = p^a$, where p be a prime number, a be a positive integer, then

$$|GL_2(\mathbf{Z}_{p^a})| = p^{4a-3}(p^2 - 1)(p - 1),$$

if $n = pq$, where p,q be the prime numbers, then

$$\left|GL_2(\mathbf{Z}_{pq})\right| = p(p^2-1)(p-1)q(q^2-1)(q-1).$$

II) Encryption

User Bob performs the following actions.

1. Presents plaintext as a sequence of *N* nonnegative integers l_1, …, l_N , where each $l_i \in \mathbf{Z}_n, i = 1,\ldots,N$ and a positive integer *N* divisible by 4. If necessary, for the latest four of these numbers using the cyclic shift.

2. Writes the first four numbers in the matrix form $m_1 = \begin{pmatrix} l_1 & l_2 \\ l_3 & l_4 \end{pmatrix} \in M_2(\mathbf{Z}_n)$, the remaining quartets writes in matrices in the same way.

3. For each of the obtained matrix $m^{(i)}, i = 1,\ldots,\frac{N}{4}$ selects session keys - the random integers r_i, t_i satisfying

$$-f(n)+2 \le r_i, t_i \le f(n)-2.$$

4. For each matrix $m^{(i)}$ receives a block of ciphertext $C^{(i)}$:

$$C^{(i)} = C_1^{(i)}, C_2^{(i)} = P_3^{-r_i} P_1^{t_i} P_3^{r_i}, m^{(i)} P_3^{-r_i} P_2^{-t_i} P_3^{r_i} \ , \ i = 1,2,\ldots,\frac{N}{4}.$$

5. Sends to Alice the ciphertext *C* as a concatenation of blocks $C^{(i)}, i = 1,\ldots,\frac{N}{4}$:

$$C = C^{(1)} \| C^{(2)} \| \ldots \| C^{\left(\frac{N}{4}\right)}.$$

III) Decryption

After receiving the ciphertext *C*, Alice, using the private key S_A , computes [16]:

$$C_2^{(i)} U_n^{-s} (C_1^{(i)})^k U_n^s = m^{(i)}, i = 1,\ldots,\frac{N}{4}.$$

As a result, Alice will due and is able to restore the original plaintext. If it refers to one matrix, then the index will be omitted.

Note that in step 4 of encryption a matrix C_2 can be replaced by

$$C_2 = P_3^{-r} P_2^{-t} P_3^{r} m P_3^{-r} P_2^{-t} P_3^{r},$$

the modification of decryption is as follows:

$$U_n^{-s} C_1^k U_n^s C_2 U_n^{-s} C_1^k U_n^s = m.$$

This encryption option is called a closed version of BMMC [16].

Example 1

1) Key generation

Alice performs the following actions.

1. Selects $n = 17^2 = 4913$.
2. Selects the words

$$W(X) = B^{-1}AB^{-1}C^{-2}ABC^{-1}ABCA^{-1}C^{-1}AC, \quad W(U) = C^3BC^{-1}AC^{-1}AC^{-1}BC^{-1}B^{-3}.$$

3. Transforms the words in matrices

$$X_n = \begin{pmatrix} 3284 & 2393 \\ 4688 & 2499 \end{pmatrix}, \; U_n = \begin{pmatrix} 349 & 4640 \\ 3115 & 3870 \end{pmatrix}.$$

4. Computes the order of the general linear group $GL_2(\mathbf{Z}_n)$:

$$f(n) = 546452934898176,$$

chooses the random integers

$$k = -108644735397888, \; s = 392065451882410, \; l = 27722.$$

5. Computes matrices

$$U^l = \begin{pmatrix} 519 & 932 \\ 2341 & 1222 \end{pmatrix}, \quad X^k = \begin{pmatrix} 3129 & 2737 \\ 3315 & 3231 \end{pmatrix}, \quad U^s = \begin{pmatrix} 113 & 812 \\ 4844 & 852 \end{pmatrix}.$$

Alice public key:

$$P_A = \left(n = 4913, P_1 = \begin{pmatrix} 3284 & 2393 \\ 4688 & 2499 \end{pmatrix}, P_2 = \begin{pmatrix} 4676 & 629 \\ 2771 & 1684 \end{pmatrix}, P_3 = \begin{pmatrix} 519 & 932 \\ 2341 & 1222 \end{pmatrix} \right).$$

Alice private key:

$$S_A = \left(U_n = \begin{pmatrix} 349 & 4640 \\ 3115 & 3870 \end{pmatrix}, \; k = -108644735397888, \; s = 392065451882410 \right).$$

2) Encryption

Bob performs the following actions.

1. From plaintext “algebra” by replacing letters in the table ASCII codes is obtained matrix (using cyclic shift the letter "*a*" to the complement of the last four):

$$m_1 = \begin{pmatrix} 97 & 108 \\ 103 & 101 \end{pmatrix}, \; m_2 = \begin{pmatrix} 98 & 114 \\ 97 & 97 \end{pmatrix}.$$

2. For matrix m_1 selects integers $r_1 = -546452685450077$, $t_1 = -546452591582313$ and for matrix m_2 selects integers $r_2 = -546452670794053$, $t_2 = 42865650$.

3. Obtains the ciphertext blocks:

$$C^{(1)} = \left(\begin{pmatrix} 4330 & 4621 \\ 4587 & 4775 \end{pmatrix}, \begin{pmatrix} 3497 & 2454 \\ 3826 & 4657 \end{pmatrix} \right), C^{(2)} = \left(\begin{pmatrix} 4761 & 452 \\ 102 & 3333 \end{pmatrix}, \begin{pmatrix} 4161 & 2052 \\ 4024 & 4806 \end{pmatrix} \right).$$

4. Ciphertext will be:

$$C = C^{(1)} \,\|\, C^{(2)}.$$

3) Decryption

Alice, using her private key, computes

$$\begin{pmatrix} 3497 & 2454 \\ 3826 & 4657 \end{pmatrix}\begin{pmatrix} 349 & 4640 \\ 3115 & 3870 \end{pmatrix}^{-s}\begin{pmatrix} 4330 & 4621 \\ 4587 & 4775 \end{pmatrix}^{k}\begin{pmatrix} 349 & 4640 \\ 3115 & 3870 \end{pmatrix}^{s} = m_1 ,$$

$$\begin{pmatrix} 4161 & 2052 \\ 4024 & 4806 \end{pmatrix}\begin{pmatrix} 349 & 4640 \\ 3115 & 3870 \end{pmatrix}^{-s}\begin{pmatrix} 4761 & 452 \\ 102 & 3333 \end{pmatrix}^{k}\begin{pmatrix} 349 & 4640 \\ 3115 & 3870 \end{pmatrix}^{s} = m_2$$

and obtains the word “algebraa”. The last letter “a” is obtained from a cyclic shift (since the number of letters is not a multiple of 4 in step 1 of encryption) and removed from the resulting text.

In this example the small number *n* is used. In fact, in cryptosystem BMMC are used no less than 64-bit numbers.

3. Diffie–Hellman protocol

Let us consider the classical Diffie-Hellman protocol [17]. So, let two users Alice and Bob distant from each other and communicate in an open channel of communication. It is assumed that at the beginning of communication they have no joint secret information, and by the end of the session they should have a shared secret key. It is believed that the attacker Mallory obtains the transmitted information between Alice and Bob.

1. Alice and Bob agree on a prime number *p* and a generator *g* of a multiplicative group $\mathbf{Z}_p^*$ of the field $\mathbf{Z}_p$ of residues modulo *p*.

2. Alice chooses a random integer *a* such that $2 \le a \le p-2,$

computes

$$M = g^a \bmod p$$

and sends *M* to Bob.

3. Bob chooses a random integer *b* such that $2 \le b \le p-2,$

computes

$$N = g^b \bmod p$$

and sends *N* to Alice.

4. Alice computes

$$K = N^a \bmod p .$$

5. Bob computes

$$K = M^b \bmod p .$$

Thus, Alice and Bob jointly formed a shared secret key *K* of the symmetric cryptosystem.

Indeed,

$$N^a \bmod p = (g^b \bmod p)^a \bmod p = g^{ba} \bmod p ,$$

$$M^b \bmod p = (g^a \bmod p)^b \bmod p = g^{ab} \bmod p$$

and as for integers $ab = ba$, then

$$N^a \bmod p = M^b \bmod p = K .$$

4. “Man in the middle” attack on the Diffie-Hellman protocol

Unfortunately, the Diffie–Hellman key exchange protocol is not recommended for use in practice because of the following well-known “man in the middle” attack on this protocol [17].

1. Alice and Bob perform step 1 of the Diffie–Hellman protocol. Available information from this point i.e. prime p and generator g becomes known Mallory , which forging protocol.

2. Alice performs step 2 of the Diffie–Hellman protocol and sends Bob the number

$$M = g^a \bmod p .$$

3. Bob performs step 3 of the Diffie–Hellman protocol and sends Alice the number

$$N = g^b \bmod p .$$

4. Mallory intercepts of M and N, computes

$$L = g^c \bmod p \text{ and } P = g^d \bmod p .$$

5. Mallory sends Alice and Bob numbers L and P, respectively.

6. Alice performs step 4 of the Diffie–Hellman protocol and computes

$$L^a \bmod p = (g^c \bmod p)^a \bmod p = g^{ca} \bmod p = K_1 .$$

7. Bob performs step 5 of the Diffie–Hellman protocol and computes

$$P^b \bmod p = (g^d \bmod p)^b \bmod p = g^{db} \bmod p = K_2 .$$

8. Mallory computes the numbers

$$M^c \bmod p = (g^a \bmod p)^c \bmod p = g^{ac} \bmod p = K_1$$

and

$$N^d \bmod p = (g^b \bmod p)^d \bmod p = g^{bd} \bmod p = K_2.$$

As a result, Alice and Bob believe that they have worked out according to the protocol the shared secret key, at the time, as there are two different keys K_1 and K_2 that are shared Alice and Mallory, Bob and Mallory, respectively. Therefore, Mallory can control the secret correspondence of Alice and Bob, without betraying itself. To this end Mallory enough, using a shared key with Alice, decrypt her messages to Bob, encrypted using this key, and read them without changing the messages, encrypt them on the shared key with Bob and send to Bob. Similarly do with the messages from Bob to Alice.

It should be noted that Mallory can compromise the Diffie-Hellman key establishment protocol and other means, if it can solve the discrete logarithm problem (that is, to find the discrete logarithms a and b of the numbers $M = g^a \bmod p$ and $N = g^b \bmod p$ respectively), then to compute ab and $K = g^{ab} \bmod p$. Or be able to solve the Diffie-Hellman problem: the known numbers $g^a \bmod p$ and $g^b \bmod p$ to find the number $g^{ab} \bmod p$.

5. Noncommutative analogue of the Diffie–Hellman protocol

We try to apply the cryptosystem BMMC for constructing noncommutative analogue of the Diffie-Hellman protocol to repel “man in the middle” attack and the threat of solving the discrete logarithm problem on a quantum computer. The use of electronic digital signature (EDS) in the Diffie-Hellman protocol only partially solves this problem because it does not protect against the potential threat of an attack on a quantum computer in view of the commutativity of the points group of an elliptic curve on a finite field used in all national standards EDS now. By the way, this fact makes the problem of development in the framework of noncommutative algebraic cryptography reliable and practical EDS on an appropriately chosen noncommutative algebraic structure .

Alice and Bob must perform the following steps of the protocol.

1. Alice and Bob agree on a positive integer n, which is not secret, the symmetric cryptosystem (AES) and the hash function (SHA-1).

2. Using BMMC, Alice and Bob generate their key pairs:

P_A, S_A and (P_B, S_B), where P_A and P_B are the public keys of Alice and Bob respectively, S_A and S_B are the private keys of Alice and Bob respectively.

Alice chooses a random matrix $m_A \in M_2(\mathbf{Z}_n)$ and Bob chooses a random matrix $m_B \in M_2(\mathbf{Z}_n)$.

3. Alice, using Bob's public key P_B encrypts the matrix m_A by closed variant of BMMC and sends the ciphertext to Bob. Bob does symmetrically.

4. Alice and Bob are using their private keys S_A and S_B restoring $m_B^{'}$ and $m_A^{'}$ respectively (here designations are used in the sense that at this step Mallory could replace the matrices m_B and m_A).

5. Shared secret key of the symmetric cryptosystem will be declared the SHA-1 reduced hash value of the bit string obtained from matrix $m = m_A + m_B$ by means of the binary representation of concatenation of matrix elements $m_{11} \| m_{12} \| m_{21} \| m_{22}$ with removed the least significant four bytes of hash value, if performed the identities of concatenations of matrix elements for the matrices m_A and $m_A^{'}$ also for the matrices m_B and $m_B^{'}$.

6. Alice with a previously agreed with Bob symmetric cryptosystem AES encrypts the matrix $m_B^{'}$ (i.e. concatenation of matrix elements), received from Bob, using her copy of the shared key, received from the matrix $m_A + m_B^{'}$. Bob, using the same symmetric cryptosystem encrypts the matrix $m_A^{'}$, received from Alice, using his copy of the shared key, received from the matrix $m_B + m_A^{'}$. If the bit length of the plaintext (or the last plaintext block) is less than 128 bits, protocol participants complement it with space characters up to 16 bytes.

Next, Alice and Bob exchange ciphertexts.

7. Each participant of the protocol decrypts the received ciphertext by his copy of a shared secret key and removes the space characters, if they are. Alice checks for identity of the concatenations of the matrices elements $m_A^{'}$ and m_A. Bob checks for identity of the concatenations of the matrices elements m_B and $m_B^{'}$.

8. If both identities are satisfied, then Alice and Bob possess a shared secret key. Otherwise, the fact that the replacement is considered established and shared secret in this session of protocol is not formed.

Note that the closed variant of BMMC need to prevent Mallory from distorting the ciphertexts. Steps 7, 8 confirm that the protocol is performed correctly, Alice and Bob are received the same secret key and replacement of keys did not happen. In fact, these steps are replaced with a digital signature to the Diffie-Hellman protocol.

Example 2

1. Alice and Bob agree on a positive integer n = 4913, symmetric cipher AES and hash function SHA-1.

2. Alice and Bob choose BMMC key pairs, e.g., the same as in example 1.

3. Alice chooses a random matrix $m_A = \begin{pmatrix} 97 & 110 \\ 102 & 106 \end{pmatrix}$, Bob chooses a random matrix $m_B = \begin{pmatrix} 100 & 102 \\ 114 & 103 \end{pmatrix}$.

4. Alice using Bob's public key encrypts matrix m_A and sends to Bob ciphertext – the pair of matrices

$$\left(\begin{pmatrix} 847 & 3737 \\ 1643 & 39 \end{pmatrix}, \begin{pmatrix} 1625 & 4407 \\ 2287 & 4034 \end{pmatrix} \right).$$

Bob using Alice's public key encrypts matrix m_B and sends to Alice ciphertext – the pair of matrices

$$\left(\begin{pmatrix} 3323 & 4497 \\ 1267 & 3338 \end{pmatrix}, \begin{pmatrix} 219 & 4352 \\ 743 & 4574 \end{pmatrix} \right)$$

.

5. Alice and Bob using their private keys restore

$$m_B^{'} = \begin{pmatrix} 100 & 102 \\ 114 & 103 \end{pmatrix},\ m_A^{'} = \begin{pmatrix} 97 & 110 \\ 102 & 106 \end{pmatrix}.$$

Alice applies a hash function SHA-1 to the concatenation of the elements of matrix $m_A + m_B^{'}$,

computes the hash value h(197212216209) and receives 20 bytes of the hash value in the hexadecimal code:

“01 d8 9e e1 0f bf 06 b9 fd 1f 90 44 4b 9b 59 61 43 cd 50 dc”.

The resulting her copy of AES 128-bit shared secret key is a reduced hash value by means of removing 4 least significant bytes:

“01 d8 9e e1 0f bf 06 b9 fd 1f 90 44 4b 9b 59 61”.

Bob applies a hash function SHA-1 to the concatenation of the elements of matrix $m_B + m_A^{'}$

and receives as a result his copy of AES 128-bit shared secret key:

“01 d8 9e e1 0f bf 06 b9 fd 1f 90 44 4b 9b 59 61”.

6. Alice using her copy of shared secret key encrypts matrix $m_B^{'}$ as plaintext complementing space characters up to 16 bytes (i.e. plaintext is "100102114103", complemented plaintext in ASCII code is

“31 30 30 31 30 32 31 31 34 31 30 33 20 20 20 20”,

ciphertext in hex code is

“ab f9 c7 cf 9d 00 94 8e 8e ca 76 04 98 24 f2 11 “).

Bob using his copy of shared secret key encrypts matrix $m_A^{'}$ in the same manner (i.e., plaintext is " 97110102106 " , complemented plaintext in ASCII code is

“39 37 31 31 30 31 30 32 31 30 36 20 20 20 20 20”,

ciphertext in hex code is

“cc bf ed 7f 23 80 06 d0 cb d8 2d a2 a3 63 99 3e”).

Next, Alice and Bob are exchanged ciphertexts.

7. Each participant decrypts received ciphertext with his copy of shared secret key. Alice gets in ASCII code

“39 37 31 31 30 31 30 32 31 30 36 20 20 20 20 20”,

removes the space characters and receives the plaintext “97110102106” – the concatenation of elements of matrix $m_A^{'}$ – and compares it with the concatenation of elements of matrix m_A - “97110102106”. Bob gets in ASCII code

“31 30 30 31 30 32 31 31 34 31 30 33 20 20 20 20”,

removes the space characters and receives the plaintext “100102114103” – the concatenation of elements of matrix $m_B^{'}$ – and compares it with the concatenation of elements of matrix m_B – “100102114103”.

8. Both concatenations of the matrices elements are identical for each of the protocol participants. Therefore, shared secret key is formed.

6. Attack of the “man in the middle” on the noncommutative analogue of the Diffie–Hellman protocol

Consider the attack, similar to the one that was implemented for the Diffie–Hellman key establishment protocol in section 4. Recall that as a result of this attack, Mallory gained control of the secret correspondence between Alice and Bob, and they had no means to determine the presence or absence of such control. We show that the noncommutative analogue, in contrast to the classical Diffie–Hellman protocol, provides such facilities.

1. Alice and Bob perform the first four steps of noncommutative analogue of the Diffie–Hellman protocol.

2. Mallory chooses the matrices m_M^A, m_M^B and encrypts them using the public keys of Alice and Bob, respectively, and then intercepted them ciphertexts sent between Alice and Bob, replace their ciphertexts.

3. Alice and Bob perform the step 5 of the protocol and receive $m'_B = m^A_M$ and $m'_A = m^B_M$ respectively.

4. Alice and Bob perform the steps 6, 7 and 8 of the protocol.

Alice's copy of shared key - it's a reduced hash value of the bit string of matrix $m_A + m^A_M$, and Bob's copy of shared key - it's a reduced hash value of the bit string of matrix $m_B + m^B_M$. Mallory knows only the second terms of Alice and Bob's shared key (in contrast to the Diffie–Hellman protocol, which he knew the numbers themselves) and therefore can only try to guess the copies, it is clear that the probability of guessing is negligible. Thus, Mallory does not have shared keys to Alice and Bob, and therefore can not control their secret correspondence. On the other hand, in the case of substitution Mallory at least one of the matrices, the probability of coincidence of copies Alice and Bob's shared keys is also negligible. Then if Alice and Bob have different keys symmetric cryptosystem AES, they can not properly decrypt ciphertexts encrypted by another key. In this case at least one of the identities of both concatenations of the matrices elements m_A and m'_A also m_B and m'_B is not performed. Thus, an attempt Mallory to attack the protocol immediately detected. At the same time, if there is no attack, both identities hold, and the shared secret key is obtained.

References

[1] Shor, P.W. 1994. Algorithms for Quantum Computation: Discrete Logarithm and Factoring. In Proceedings of the IEEE 35th Annual Simposium on Foundations of Computer Science, 124-134.

[2] Anshel, I., Anshel, M., and Goldfeld, D. 1999. An Algebraic Method for Public Key Cryptography. Math. Res. Letters, 6, 287–291.

[3] Baumslag, G., Fine, B., and Xu, X. 2006. Cryptosystems Using Linear Groups. Applicable Algebra in Engineering, Communication and Computing, 17, 205–217.

[4] Baumslag, G., Biryukhov, Y., Fine, B., and Rosenberger, G. 2008. Some Cryptoprimitives for Noncommutative Algebraic Cryptography. In Algebra and Discrete Mathematics, vol. 1, Aspects of Infinite Groups.

[5] Baumslag, G., Camps, T., Fine, B., Rosenberger, G., and Xu, X. 2006. Designing Key Transport Protocols Using Combinatorial Group Theory. Contemp. Math., 418, 35-43.

[6] Baumslag, G., Fine, B., and Xu, X. 2007. A Proposed Public Key Cryptosystem Using the Modular Group . Contemp. Math., 421, 35–44.

[7] Ko, K. H., Lee, J., Cheon, J.H., Han, J.W., Kang, J., and Park, C. 2000. New Public Key Cryptosystem Using Braid Groups. In Advances in Cryptology, CRYPTO 2000, Lect. Notes in Computer Science, v.1880, 166–183.

[8] Koblitz, N. 1998. Algebraic Methods of Cryptography. Springer.

[9] Yamamura, A. 1998. Public Key Cryptosystems Using the Modular Group. In Lect. Notes in Computer Science, v.1431, 203–216.

[10] Grigoriev, D., and Ponomarenko, I. 2005. Homomorphic Public Key Cryptosystems over Groups and Rings. Quaderni di Matematica.

[11] Paeng, S.-H., Ha, K.-C., Kim, J.N., Chee, S., and Park, C. 2001. New Public Key Cryptosystem Using Finite Non-Abelian Groups. In Proceedings of the Crypto 2001, 470–485.

[12] Rososhek, S.K. 2008. Cryptosystems in Automorphism Groups of Group Rings of Abelian Groups. Journal of Mathematical Sciences, 154, 386–391.

[13] Gribov, A.N., Zolotykh, P.A., and Mikhalev, A.V. 2010. A construction of algebraic cryptosystem over the quasigroup ring. Mathematical Aspects of Cryptography, 1, 23–32 (Russian).

[14] Hall, C., Goldberg, I., and Schneier, B. 1999. Reaction attacks Against Several Public Key Cryptosystems, In Proceedings of Information and Communication security, ICICS 99, 2–12.

[15] Steinwandt, R. 2001. Loopholes in Two Public Key Cryptosystems using the Modular Group. In Lect. Notes in Computer Science, v. 1992, 180–189.

[16] Rososhek, S.K. 2013. New Practical Algebraic Public Key Cryptosystem and Some Related Algebraic and Computational Aspects. Applied Math., 4, 1043–1049.

[17] Smart, N. 2003. Cryptography: Introduction. McGraw-Hill.