Proceedings of the Workshop on Foundations of Informatics
FOI-2015, August 24-29, 2015, Chisinau, Republic of Moldova

On some applications of quasigroups in cryptography

N.A. Moldovyan A.V. Shcherbacov V.A. Shcherbacov

Abstract

In the paper we present based on quasigroups new deniable encryption method, relatively fast stream cipher and generalisation of El-Gamal scheme.

Keywords: cryptology, quasigroup, algorithm, stream, deniable, encryption, El-Gamal scheme

1 Deniable-encryption mode for block ciphers

Deniable encryption (DE) is a method for generating ciphertexts that can be alternatively decrypted providing security against the so called coercive attacks [3] for which it is assumed that after ciphertext has been sent the adversary has possibility to force both the sender and the receiver to open the plaintext corresponding to the ciphertext and the encryption key. In the case of block ciphering the DE can be provided with simultaneous encryption of the secret and fake messages using the secret and fake keys, correspondingly. While being coerced the sender and receiver of the ciphertext open the fake key and fake message and declare they have used the probabilistic encryption [4]. Earlier in paper [5] it had been proposed a method for simultaneous encryption of two messages based on solving a system of two linear equations. In this section we propose design of the DE mode for using block ciphers, which is based on the mentioned method.

Definition 1. *Binary groupoid* $(G, \circ)$ *is isotopic image of a binary groupoid* $(G, \cdot)$*, if there exist permutations* α, β, γ *of the set* G *such that* $x \circ y = \gamma^{-1}(\alpha x \cdot \beta y)$ *[1].*

Suppose E_V be a block encryption algorithm with n-bit input data block and the value used as encryption key. All existing n-bit data blocks can be considered as elements of some quasigroup with the operation $*$ defined as follows:

$$K * i = E_V(K \oplus E_V(i)),$$

where $\oplus$ is the XOR operartion; K and i are n-bit vectors. This quasigroup is isotope of the group $(G, \oplus)$, where G is the set of all n-bit vectors. Here E_V is a permutation of the symmetric group S_G.

Evidently, for all possible values i and $Q \neq K$ we have

$$E_V(Q \oplus E_V(i)) \neq E_V(K \oplus E_V(i)). \tag{1}$$

Using this property of the quasigroup one can define simultaneous encryption of two different messages $T = (t_1, t_2, \ldots, t_i, \ldots, t_z)$ and $M = (m_1, m_2, \ldots, m_i, \ldots, m_z)$, where $z < 2^n$; t_i and m_i are n-bit data blocks, as generation of the single ciphertext $C = (c_1, c_2, \ldots, c_i, \ldots, c_z)$ containing $(2n)$-bit ciphertext blocks $c_i = (c'_i, c''_i)$, where c'_i and c''_i are n-bit values, computed from the following system of equations in the field $GF(2^n)$:

$$\begin{cases} c'_i + A_i c''_i \equiv B_i + m_i \bmod \eta(x) \\ c'_i + G_i c''_i \equiv H_i + t_i \bmod \eta(x), \end{cases} \tag{2}$$

where $\eta(x)$ is some specified irreducible binary polynomial of the degree n; the n-bit values A_i, B_i, G_i, and H_i are computed using the random n-bit initialization vector V (this value is not secret) as follows:

$$A_i = E_V(K \oplus E_V(i)); G_i = E_V(Q \oplus E_V(i));$$
$$B_i = E_K(A_i); H_i = E_Q(G_i).$$

While solving (1) the values A_i, B_i, G_i, and H_i are considered as binary polynomials of the degree $s < n$. Due to condition (1) the system (2) always has the single solution, therefore the proposed deniable-encryption procedure is defined correctly. Let us agree that the secret message (key) is the value T (Q) and the fake message (key) is the value M (K). If the coercer forces the sender and receiver of the secret message T to open the ciphertext C and the encryption key, then

they open the fake key K and the fake message M and declare using the probabilistic block-encryption mode implemented with the block cipher E. In terms of paper [4] the declared encryption algorithm is called the associated encryption algorithm.

In the case of the proposed deniable-encryption method the last algorithm is described as consecutive probabilistic encryption of the data blocks m_i for each value $i = 1, 2, \ldots, z$ performing the following steps:

1. Generate a random initialization vector V and compute the values $A_i = E_V(K \oplus E_V(i))$ and $B_i = E_K(A_i)$.

2. Generate a random binary polynomial $\rho_i(x)$ of the degree $s < n$.

3. Compute the unknowns c_i' and c_i'' from the following system of equations in $GF(2^n)$:

$$\begin{cases} c_i' + A_i c_i'' \equiv B_i + m_i \bmod \eta(x) \\ c_i' + \rho_i c_i'' \equiv 1 \bmod \eta(x), \end{cases} \tag{3}$$

Evidently, for some sequence of the values $\rho_1(x), \rho_2(x), \ldots, \rho_z(x)$ the message M is transformed with the key K into the given ciphertext C.

To distinguish the use of the deniable encryption with the system (2) from the probabilistic encryption with the system (3) the potential coercive attacker should compute the key Q. The last problem is computationally difficult, if E is a secure block cipher, for example, AES [7] with 128-bit key and $n = 128$. Restoring the secret message from the ciphertext is performed as decryption of each ciphertext block $c_i = (c_i', c_i')$, $i = 1, 2, \ldots, z$, as follows:

1. Using the secret key Q compute the values $G_i = E_V(Q \oplus E_V(i))$ and $H_i = E_Q(G_i)$.

2. Compute the plaintext data block $t_i = c_i' + G_i c_i'' - H_i \bmod \eta(x)$.

The fake decryption of the ciphertext is as follows ($i = 1, 2, \ldots, z$):

1. Using the fake key K compute the values $A_i = E_V(K \oplus E_V(i))$ and $B_i = E_K(A_i)$.

2. Compute the plaintext data block $m_i = c_i' + A_i c_i'' - B_i \bmod \eta(x)$.

2 Stream cipher on base of binary quasigroups

Here we give more detailed description of algorithm which was proposed in [10]. This algorithm simultaneously uses two cryptographical procedures: enciphering using generalisation of Markovski stream algorithm [11] and enciphering using a system of orthogonal operations.

We also give some realisation of this algorithm on base of T-quasigroups, more precise, on the base of medial quasigroups. Necessary information about quasigroups and some its applications in cryptography can be found in [1, 8, 10].

Below we denote the action of the left (right, middle) translation in the power a of a binary quasigroup (Q, g_1) on the element u_1 by the symbol ${}_{g_1}T^a_{l_1}(u_1)$. And so on. Here l_1 means leader element. See [8, 10, 11] for details.

Algorithm 1. *Enciphering. Initially we have plaintext* $u_1, u_2, \ldots, u_6$.

$$
\begin{aligned}
&\textit{Step } 1.\\
&{}_{g_1}T^a_{l_1}(u_1) = v_1\\
&{}_{g_2}T^b_{l_2}(u_2) = v_2\\
&F^c_1(v_1, v_2) = (v'_1, v'_2)\\
&\textit{Step } 2.\\
&{}_{g_3}T^d_{v'_1}(u_3) = v_3\\
&{}_{g_4}T^e_{v'_2}(u_4) = v_4\\
&F^f_2(v_3, v_4) = (v'_3, v'_4)\\
&\textit{Step } 3.\\
&{}_{g_5}T^g_{v'_3}(u_5) = v_5\\
&{}_{g_6}T^h_{v'_4}(u_6) = v_6\\
&F^i_3(v_5, v_6) = (v'_5, v'_6).
\end{aligned}
\tag{4}
$$

We obtain ciphertext $v'_1, v'_2, \ldots, v'_6$.

Deciphering. Initially we have ciphertext $v'_1, v'_2, \ldots, v'_6$.

$$
\begin{array}{l}
\textit{Step } 1. \\
F_1^{-c}(v'_1, v'_2) = (v_1, v_2) \\
{}_{g_1}T_{l_1}^{-a}(v_1) = u_1 \\
{}_{g_2}T_{l_2}^{-b}(v_2) = u_2 \\
\textit{Step } 2. \\
F_2^{-f}(v'_3, v'_4) = (v_3, v_4) \\
{}_{g_3}T_{v'_1}^{-d}(v_3) = u_3 \\
{}_{g_4}T_{v'_2}^{-e}(v_4) = u_4 \\
\textit{Step } 3. \\
F_3^{-i}(v'_5, v'_6) = (v_5, v_6) \\
{}_{g_5}T_{v'_3}^{-g}(v_5) = u_5 \\
{}_{g_6}T_{v'_4}^{-h}(v_6) = u_6
\end{array}
\tag{5}
$$

We obtain plaintext $u_1, u_2, \ldots, u_6$.

From Algorithm 1 we obtain classical Markovski algorithm, if we take only one quasigroup, one kind of quasigroup translations (left translations) any of which is taken in power = 1, and, finally, if system of orthogonal operations (crypto-procedure F) is not used. Some generalisations of Algorithm 1 are given in [12].

3 T-quasigroup based stream cipher

We give a numerical example of encryption Algorithm 1 based on T-quasigroups (more exactly, on medial quasigroups) [12]. Notice that the number 257 is prime. Form of parastrophes of T-quasigroups, for example, of quasigroup $(A, \overset{(13)}{*})$ can be found in [12], [6, p. 39].

Example 1. *Take the cyclic group* $(Z_{257}, +) = (A, +)$.

1. *Define T-quasigroup* $(A, *)$ *with the form* $x * y = 2 \cdot x + 131 \cdot y + 3$ *with a leader element* l*, say,* $l = 17$*. Denote the mapping* $x \mapsto x * l$ *by the letter* g_1*, i.e.* $g_1(x) = x * l$ *for all* $x \in A$*.*

   In order to find the mapping g_1^{-1} *we find the form of operation* $\overset{(13)}{*}$*. We have* $x \overset{(13)}{*} y = 129 \cdot x + 63 \cdot y + 127$*,* $f^{-1}x = x \overset{(13)}{*} l$*. Then* $g_1^{-1}(g_1(x)) = g_1^{-1}(x * l) = (x * l) \overset{(13)}{*} l = x$*.*

   In some sense quasigroup $(A, \overset{(13)}{*})$ *is the "right inverse quasigroup" to quasigroup* $(A, *)$*. Notice that from results of article [6, Theorem 16] it follows that* $(A, *) \perp (A, \overset{(13)}{*})$*.*

2. *Define T-quasigroup* $(A, \circ)$ *with the form* $x \circ y = 10 \cdot x + 81 \cdot y + 53$ *with a leader element* l*, say,* $l = 71$*. Denote the mapping* $x \mapsto l * x$ *by the letter* g_2*, i.e.* $g_2(x) = l \circ x$ *for all* $x \in A$*.*

   In order to find the mapping g_2^{-1} *we find the form of operation* $\overset{(23)}{\circ}$*. We have* $x \overset{(23)}{\circ} y = 149 \cdot x + 165 \cdot y + 250$*.*

3. *Define a system of two parastroph orthogonal T-quasigroups* $(A, \cdot)$ *and* $(A, \overset{(23)}{\cdot})$ *in the following way*

$$\begin{cases} x \cdot y = 3 \cdot x + 5 \cdot y + 6 \\ x \overset{(23)}{\cdot} y = 205 \cdot x + 103 \cdot y + 153. \end{cases}$$

   Denote quasigroup system $(A, \cdot, \overset{(23)}{\cdot})$ *by* $F(x, y)$*, since this system is a function of two variables.*

   In order to find the mapping $F^{-1}(x, y)$ *we solve the system of linear equations*

$$\begin{cases} 3 \cdot x + 5 \cdot y + 6 = a \\ 205 \cdot x + 103 \cdot y + 153 = b. \end{cases}$$

   We have $\Delta = 55$*,* $1/\Delta = 243$*,* $x = 100 \cdot a + 70 \cdot b + 255$*,* $y = 43 \cdot a + 215 \cdot b$*. Therefore we have, if* $F(x, y) = (a, b)$*, then* $F^{-1}(a, b) =$

$(100 \cdot a + 70 \cdot b + 255, 43 \cdot a + 215 \cdot b)$, *i.e.*

$$\begin{cases} x = 100 \cdot a + 70 \cdot b + 255 \\ y = 43 \cdot a + 215 \cdot b. \end{cases}$$

We have defined the mappings g_1, g_2, F *and now we can use them in Algorithm 1.*

Let 212; 17; 65; 117 *be a plaintext. We take the following values in formula (4):* $a = b = d = e = f = 1; c = 2$. *Below we use Gothic font to distinguish leader elements, i.e., the numbers* $\mathfrak{17}$ *and* $\mathfrak{71}$ *are leader elements. Then*

Step 1.

$g_1(212) = 212 * \mathfrak{17} = 2 \cdot 212 + 131 \cdot 17 + 3 = 84$

$g_2(17) = \mathfrak{71} \circ 17 = 10 \cdot 71 + 81 \cdot 17 + 53 = 84$

$F(84; 84) = (3 \cdot 84 + 5 \cdot 84 + 6; 205 \cdot 84 + 103 \cdot 84 + 153) = (164; 68)$

$F(164; 68) = (3 \cdot 164 + 5 \cdot 68 + 6; 205 \cdot 164 + 103 \cdot 68 + 153) =$ ***(67; 171)***

Step 2.

$g_1(65) = 65 * 67 = 2 \cdot 65 + 131 \cdot 67 + 3 = 172$

$g_2(117) = 171 \circ 117 = 10 \cdot 171 + 81 \cdot 117 + 53 = 189$

$F(172; 189) = (3 \cdot 172 + 5 \cdot 189 + 6; 205 \cdot 172 + 103 \cdot 189 + 153) =$ ***(182; 139)***

We obtain the following ciphertext 67; 171; 182; 139.

For deciphering we use formula (5).

Step 1.

$F^{-1}(67; 171) = (100 \cdot 67 + 70 \cdot 171 + 255, 43 \cdot 67 + 215 \cdot 171) = (164; 68)$

$F^{-1}(164; 68) = (100 \cdot 164 + 70 \cdot 68 + 255, 43 \cdot 164 + 215 \cdot 68) = (84; 84)$

$g_1^{-1}(84) = 84 \overset{(13)}{*} 17 = 129 \cdot 84 + 63 \cdot 17 + 127 =$ ***212***

$g_2^{-1}(84) = 71 \overset{(23)}{\circ} 84 = 149 \cdot 71 + 165 \cdot 84 + 250 =$ ***17***

Step 2.

$F^{-1}(182; 139) = (100 \cdot 182 + 70 \cdot 139 + 255, 43 \cdot 182 + 215 \cdot 139) = (172; 189)$

$g_1^{-1}(172) = 172 \overset{(13)}{*} 67 = 129 \cdot 172 + 63 \cdot 67 + 127 =$ ***65***

$$g_2^{-1}(189) = 171 \overset{(23)}{\circ} 189 = 149 \cdot 171 + 165 \cdot 189 + 250 = \boldsymbol{117}$$

A program using freeware version of programming language Pascal was developed. Experiments demonstrate that encoding-decoding is executed sufficiently fast.

Remark 1. *Proper binary groupoids are more preferable than linear quasigroups by construction of the mapping $F(x, y)$ in order to make encryption more safe, but in this case decryption may be slower than in linear quasigroup case and definition of these groupoids needs more computer (or some other device) memory. The same remark is true for the choice of the function g. Maybe a golden mean in this choice problem is to use linear quasigroups over non-abelian, especially simple, groups.*

Remark 2. *In this cipher there exists a possibility of protection against standard statistical attack. For this scope it is possible to denote more often used letters or pair of letters by more than one integer or by more than one pair of integers.*

4 De-symmetrisation of Markovski algorithm

We give an analogue of El Gamal encryption system based on Markovski algorithm.

Let (Q, f) be a binary quasigroup and $T = (\alpha, \beta, \gamma)$ be its isotopy.

Alices keys are as follows:

Public Key is (Q, f), T, $T^{(m,n,k)} = (\alpha^m, \beta^n, \gamma^k)$, $m, n, k \in \mathbb{N}$, and Markovski algorithm.

Private Key m, n, k.

Encryption

To send a message $b \in (Q, f)$ Bob computes $T^{(r,s,t)}$, $T^{(mr,ns,kt)}$ for a random $r, s, t \in \mathbb{N}$ and $(T^{(mr,ns,kt)}(Q, f))$.

The ciphertext is $(T^{(r,s,t)}, T^{(mr,ns,kt)}(Q, f), (T^{(mr,ns,kt)}(Q, f))b)$.

To obtain $(T^{(mr,ns,kt)}(Q, f))b$ Bob uses Markovski algorithm which is known to Alice.

Decryption

Alice knows m, n, k, so if she receives the ciphertext

$$(T^{(r,s,t)}, T^{(mr,ns,kt)}(Q,f), (T^{(mr,ns,kt)}(Q,f))b),$$

she computes $T^{(-rm,-ns,-kt)}$ from $T^{(r,s,t)}$ and then (Q,f), further she computes $(Q,f)^{-1}$ and, finally, she computes b.

In this algorithm it can also be used isostrophy [9] instead of isotopy, Algorithm 1 instead of Markovski algorithm, n-ary ($n > 2$) quasigroups [2, 10] instead of binary quasigroups.

References

[1] V.D. Belousov. *Foundations of the Theory of Quasigroups and Loops.* Nauka, Moscow, 1967. (in Russian).

[2] V.D. Belousov. *n-Ary Quasigroups.* Stiintsa, Kishinev, 1971. (in Russian).

[3] R. Canetti, C. Dwork, M. Naor, R.Ostrovsky. *Deniable Encryption.* Proceedings Advances in Cryptology CRYPTO 1997, Lectute Notes in Computer Science, 1294:90–104, 1997.

[4] A.A. Moldovyan, N.A. Moldovyan. *Practical method for bi-deniable public-key encryption.* Quasigroups and related systems, 22:277–282, 2014.

[5] A.A. Moldovyan, N.A. Moldovyan, V. A. Shcherbacov. *Bi-deniable public-key encryption protocol secure against active coercive adversary.* Buletinul Academiei de Stiinte a Republicii Moldova. Matematica, (3):23–29, 2014.

[6] G.L. Mullen, V.A. Shcherbacov. *On orthogonality of binary operations and squares.* Bul. Acad. Stiinte Repub. Mold., Mat., (2):3–42, 2005.

[7] J. Pieprzyk, Th. Hardjono, J. Seberry. *Fundumentals of Computer Security.* Springer-Verlag, Berlin, 2003.

[8] V.A. Shcherbacov. *Elements of quasigroup theory and some its applications in code theory,* 2003. links: www.karlin.mff.cuni.cz/drapal/speccurs.pdf; http://de.wikipedia.org/wiki/Quasigruppe.

[9] V.A. Shcherbacov. *On the structure of left and right F-, SM- and E-quasigroups.* J. Gen. Lie Theory Appl., 3(3):197–259, 2009.

[10] V.A. Shcherbacov. *Quasigroups in cryptology.* Comput. Sci. J. Moldova, 17(2):193–228, 2009.

[11] V.A. Shcherbacov, N.A. Moldovyan. *About one cryptoalgorithm.* In Proceedings of the Third Conference of Mathematical Society of the Republic of Moldova dedicated to the 50th anniversary of the foundation of the Institute of Mathematics and Computer Science, August 19-23, 2014, Chisinau, pp. 158–161, Chisinau, 2014. Institute of Mathematics and Computer Science.

[12] Victor Shcherbacov. *Quasigroup based crypto-algorithms,* 2012. arXiv:1201.3016.

N. A. Moldovyan[1], A. V. Shcherbacov[2],
V. A. Shcherbacov[3],

Received July 15, 2015

[1] Professor, St. Petersburg Institute for Informatics and Automation of Russian Academy of Sciences
14 Liniya, 39, St.Petersburg, 199178
Russia
E–mail: nmold@mail.ru

[2] M.Sc., Theoretical Lyceum "C. Sibirschi"
Lech Kaczyski str. 4, MD-2028, Chişinău
Moldova
E–mail: admin@sibirsky.org

[3] Dr., Institute of Mathematics and Computer Science
Academy of Sciences of Moldova Academiei str. 5, MD−2028 Chişinău
Moldova
Email: scerb@math.md