NOVI SAD J. MATH.
VOL. 30, NO. 2, 2000, 43-49

Proceedings of the TARA 2000 Conference
Novi Sad, Yugoslavia, September 6-7, 2000

SECURE TWO-WAY ON-LINE COMMUNICATION BY USING QUASIGROUP ENCIPHERING WITH ALMOST PUBLIC KEY

S. Markovski[1], D. Gligoroski[1], B. Stojčevska[1]

Abstract. Security is one of the issues in trend in networking domain. To obtain a reliable method for providing secure communication is one of the basic tasks in this area. We implemented a solution by using a Quasigroup Enciphering method with almost public key as a new concept explained in detail in this paper. This Quasigroup Enciphering method is based on using quasigroups for defining suitable encryption and decryption functions. The security properties of these functions make them suitable for this application. We developed a program for on-line chat using these functions implemented in C language.

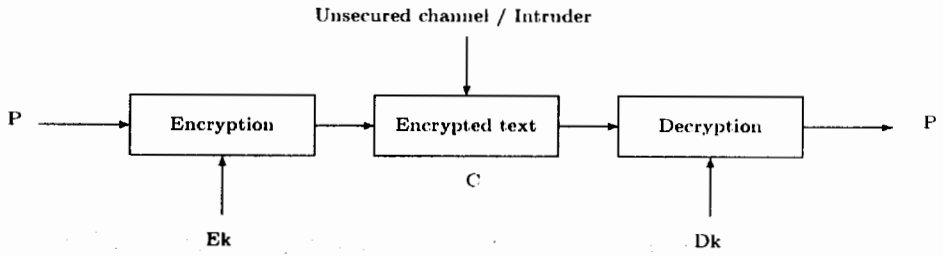
AMS Mathematics Subject Classification (1991): 94A60, 20N05, 68P30

Key words and phrases: encryption and decryption functions, on-line communication, quasigroup, stream cipher, quasigroup string processing

1. Introduction

The basic task of cryptography is obtaining secrecy, authentication, authorization and integrity of data which is sent over an unsecured communication channel or stored on publicly available medium. For that purpose a cryptographic algorithm has to be used. Formally, a set of cryptographic algorithms or a cipher is defined as a set of mathematical functions used for crypting and decrypting messages. The communication process in a model where there is an exchange of information with the possibility of intrusion can be illustrated by the following diagram:

[1] "St. Cyril and Methodius" University, Faculty of Sciences, Institute of Informatics, P. O. Box 162, Skopje, Republic of Macedonia, e-mail: {smile,danilo,biljanas}@pmf.ukim.edu.mk



- The sender sends plain text P to the receiver.
- Before the data is sent through the unsecured communication channel it is transformed to cipher text C.
- Ek is an encryption function with key k and it transforms P to C.
- Dk is a decryption function with key k and it does the reversal transformation from C to P.
- There is always a premise that an intruder can listen to the cipher text C, that is why the channel is considered as unsecured. Clearly, the following relation must hold: $Dk(Ek(P)) = P$.

Secret key algorithms (symmetric algorithms) use the same key k both for encryption and for decryption. The privacy is based on the secrecy of the key. On the other hand, public key algorithms use two keys: the first is public and is used for encryption and the second is private key and is used for decryption, and it can not be derived from the public key. Symmetric ciphers are further classified in two subcategories: block ciphers and stream ciphers. Block ciphers operate on blocks of data and stream ciphers do the ciphering on a stream of data.

In Section 2 we define a stream cipher with an almost public key, based on the quasigroup method ([3], [4], [5]). Namely, a part of the key (the quasigroups) can be public, still obtaining sufficient security. The security of the method is considered in Section 3. The software implementation is explained in Section 4.

2. Quasigroup encryption with almost public key

A groupoid $(Q, *)$ is said to be a quasigroup iff

$$(\forall u, v \in Q)(\exists! x, y \in Q)(u * x = v \ \& \ y * u = v).$$

This implies:

1) $x * y = x * z \ \vee \ y * x = z * x \implies y = z$,

2) the equations $a * x = b$, $y * a = b$ have unique solutions x, y for each $a, b \in Q$.

Given a quasigroup $(Q, *)$ a new quasigroup operation $\backslash$ on the set Q can be derived by:

$$x * y = z \iff y = x \backslash z.$$

Then the algebra $(Q, *, \backslash)$ satisfies the identities

$$x \backslash (x * y) = y, \quad x * (x \backslash y) = y. \tag{1}$$

An alphabet is a finite set A, and A^+ is the set of all nonempty words (i.e. finite strings) formed by the elements of A. The elements of A^+ will be denoted by $a_1 a_2 \dots a_n$, $a_i \in A$. Let $*$ be a quasigroup operation on the set A. For each $l \in A$ we define two functions $e_l, d_l : A^+ \longrightarrow A^+$ as follows. Let $a_i \in A$. Then:

$$e_l(a_1 a_2 \dots a_n) = b_1 b_2 \dots b_n \iff$$
$$\iff b_1 = l * a_1,\ b_2 = b_1 * a_2, \dots,\ b_n = b_{n-1} * a_n,$$

$$d_l(a_1 a_2 \dots a_n) = c_1 c_2 \dots c_n \iff$$
$$\iff c_1 = l * a_1,\ c_2 = a_1 * a_2, \dots, c_n = a_{n-1} * a_n.$$

The functions e_l, d_l are called e- and d-transformations of A^+ based on the operation $*$ with leader l.

Several quasigroup operations can be defined on the set A and let $*_1$, $*_2$, ..., $*_k$ be a sequence of (not necessarily distinct) such operations. We may also choose leaders l_1, l_2, ..., $l_k \in A$ (not necessarily distinct), and then the compositions of mappings

$$E_k = E_{l_1 \dots l_k} = e_{l_1} \circ e_{l_2} \circ \dots \circ e_{l_k}$$

$$D_k = D_{l_1 \dots l_k} = d_{l_1} \circ d_{l_2} \circ \dots \circ d_{l_k}$$

are said to be E- and D-transformations of A^+ respectively. The most general kind of transformations of A^+ are mixed compositions of e- and d-transformations. Namely, let t_{l_i} denote either e- or d-transformation based on $*_i$ with leaders l_i. The composition of mappings

$$T_k = t_{l_1} \circ t_{l_2} \circ \dots \circ t_{l_k}$$

is said to be a T-transformation of A^+.

The functions E_k, D_k and T_k have several properties which are used in this application.

The operations $*$ and $\backslash$ can be used for definitions of encryption and decryption functions, according to the properties of the kind given by (1). At first, consider the operations $*$ and $\backslash$ on an alphabet A, and let $\alpha = a_1 a_2 \dots a_n \in A^+$, $a_i \in A$, $l = a_0 = b_0 \in A$. Let the transformation function e_l be based on the

operation $*$ and d_l on $\backslash$. So, if $\beta = e_l(\alpha) = b_1 b_2 \dots b_n$ and $\gamma = d_l(\alpha) = c_1 c_2 \dots c_n$ then we have

$$b_i = b_{i-1} * a_i, \quad c_i = a_{i-1} \backslash a_i, \quad i = 1, \dots, n. \tag{1}$$

Proposition 1. *$e_l(d_l(\alpha)) = \alpha = d_l(e_l(\alpha))$ for each $\alpha \in A^+$, and $e_l^{-1} = d_l$, $d_l^{-1} = e_l$.* □

Corollary 1. *The transformation functions E_k, D_k and T_k are permutations on A^+.* □

Let us have quasigroup operations $*_i$, leaders l_i and let $\backslash_i$ be associated to $*_i$. Then the inverse of $E_k = E_{l_1 \dots l_k} = e_{l_1} \circ \dots \circ e_{l_k}$, where e_{l_i} are based on $*_i$, is $E_k^{-1} = D_{l_k \dots l_1} = d_{l_k} \circ \dots \circ d_{l_1}$, where d_{l_i} are based on $\backslash_i$.

The pairs of functions

$$(E_{l_1 \dots l_k}, D_{l_k \dots l_1}) \quad \text{and} \quad (D_{l_k \dots l_1}, E_{l_1 \dots l_k})$$

can be considered as a pair of an encryption and a decryption function for strings on an alphabet A.

Considering a T-transformation $T_k = t_{l_1} \circ \dots \circ t_{l_k}$, where $t \in \{e, d\}$, the inverse of T_k is $T_k^{-1} = t'_{l_k} \circ \dots \circ t'_{l_1}$, where $t'_i = t_i^{-1}$ is the inverse of t_i. Now, (T_k, T_k^{-1}) and (T_k^{-1}, T_k) can be considered as pairs of an encryption and a decryption function as well.

We can define different ciphers corresponding to the above encryption and decryption functions. Here we define an encryption algorithm which has a part that can be public. The encryption function is T_k and the decryption function is T_k^{-1}. The key consists of three sequences:

- a sequence of (not necessarily distinct) quasigroups operations $*_1, \dots, *_r$,
- a sequence of (not necessarily distinct) leaders $l_1, l_2, \dots, l_k$ and
- a sequence of e- and d-transformations $t_{p1}, t_{p2}, \dots, t_{pk}$ where $t \in \{e, d\}$ and $p1, \dots, pk \in \{1, 2, \dots, r\}$, such that t_{pi} is pi-based transformation with leader l_i, for each $i = 1, 2, \dots, k$.

Then $T_k = t_{p1} \circ t_{p2} \circ \dots \circ t_{pk}$. Clearly, we can take that all of the sequences are secret, in such a way obtaining a private key. But we prefer to take as secret only the sequences of leaders and of e- and d-transformations, leaving the quasigroup operations public. (That is why we used the term 'almost public key'.)

3. The security of the system

In this section we state first the main theorems for the security of a defined cipher. The theorems are in details proved in [4] and [5].

Assume that $b_1 b_2 \dots b_k$, $c_1 c_2 \dots c_k \in A^+$ and $l_1, l_2 \in A$ are given. We are trying to find all quasigroup operations $*_1, *_2$ on A such that

$$E_{l_2, l_1}(b_1 b_2 \dots b_k) = c_1 c_2 \dots c_k. \tag{2}$$

The equation (2) implies that we have to find the elements $x_1, x_2, \ldots, x_k \in A$ and quasigroup operations $*_1, *_2$ on A such that

$$(3) \quad l_1 *_1 b_1 = x_1, \;\; x_1 *_1 b_2 = x_2, \;\; x_2 *_1 b_3 = x_3, \;\; \ldots, \;\; x_{k-1} *_1 b_k = x_k,$$

$$(4) \quad l_2 *_2 x_1 = c_1, \;\; c_1 *_2 x_2 = c_2, \;\; c_2 *_2 x_3 = c_3, \;\; \ldots, \;\; c_{k-1} *_2 x_k = c_k.$$

Theorem 1. ([4]) *For finding all pairs $(*_1, *_2)$ of quasigroup operations on A such that (3) and (4) are satisfied, when k is much greater than $|A|$, one needs to make at least as many trials as there are quasigroup operations on A.* □

When we have n (> 2) operations one needs to make as many trials as there are $(n-1)$-tuples of quasigroup operations on A.

The above properties give assurance against brute force attacks, since the number of quasigroups of order n is at least $n!(n-1)!\ldots 2!1!$ ([1]). Thus, there are more than 10^{90} quasigroups of order 16, and more than 10^{58000} of order 256. Just for comparative purposes, the total number of electrons in the universe is supposed to be not more then 10^{78} ([6]).

The method can be made secure against statistical kinds of attacks as well, which follows from the next theorem.

Theorem 2. *Consider an arbitrary string $\alpha = a_1 a_2 \ldots a_n \in A^+$ and let $\beta = E_k(\alpha)$, $\gamma = D_k(\alpha)$.*

(a) ([4]) *If n is large enough integer then, for each l : $1 \leq l \leq k$, the distribution of substrings of β of length l is uniform. (We note that for $l > k$ the distribution of substrings of β of length l may not be uniform.)*

(b) ([5]) *If n and k are large enough, then the distribution of substrings of γ of a fixed length l ($l \geq 1$) is uniform.* □

The above theorems guarantee the security when a private key is considered. In the case when we have almost public key, the security is based on the sequence of leaders $l_1, l_2, \ldots, l_k$ and of the sequence of e- and d-transformations $t_{p1}, t_{p2}, \ldots, t_{pk}$, since the quasigroups are public. If the alphabet A has q letters then we can form q^k different sequences of leaders, and the number of sequences $t_{p1}, t_{p2}, \ldots, t_{pk}$ is $(2r)^k$. Then the number of all almost public keys is $(2rq)^k$, which means that a satisfactory security can be obtained for relatively small values of q, r and k. Thus, for the alphabet ASCII and $r = 2$, $k = 16$ we have $(2 * 2 * 256)^{16} = 1024^{16} = 2^{160}$ different keys.

4. Application

We used the Quasigroup Enciphering method in a program for on-line chat over the Internet. We used Ytalk 3.0.2 which is public and available for distribution and editing. The software is written in C and can run on UNIX platforms.

The end users have access to a UNIX host and run a copy of the program. After negotiating about establishing the connection the users are able to send and receive text data among each other.

Since the Quasigroup Enciphering method is suitable for stream encryption, we implemented such a module and added it in the Ytalk program. The stream of text first goes through this module, here it is encrypted and then is sent. Similarly, the encrypted stream, which is received on the other side, is first decoded and then sent to be displayed on the user's screen.

In our application we use a T_k transformation build up over two quasigroups of order 128. Namely, since the last 128 ASCII characters are used by the Ytalk protocol for control purposes, we have to consider only quasigroups on the set of the first 128 ASCII. So, from now on, we assume that the order of the considered quasigroups is 128. The space needed for storing two quasigroups is 32K and every user can have its own pair of quasigroups kept in some public directory. Anyone interested in secure communication can freely download the file and save it in the directory where he keeps the executable Ytalk file. Then, both users should exchange only a secret password in order to be able to communicate. The problem of distribution of a secret password trough unsecured communication lines is out of the scope of this article but can be satisfactory solved, for example using some public key cipher [9]. We define the secret password as a character string from which we obtain the sequence of leaders $l_1, l_2, \ldots, l_k$ and the sequence of e- and d-transformations $t_{p1}, t_{p2}, \ldots, t_{pk}$.

-The length of the password determines the value k.

-The sequence of leaders $l_1, l_2, \ldots, l_k$ is actually the string of characters of the password and their values can be chosen from the set of the characters that can be input from the terminal keyboard.

-We use the first k bits $b_1, b_2, \ldots, b_k$ and starting from right to left the last k bits $b'_1, b'_2, \ldots, b'_k$ from the binary representation of the password to determine the $t_{p1}, t_{p2}, \ldots, t_{pk}$ string in the following manner: if $b_i = 1$ then $t = c$ else $t = d$ and if $b'_i = 1$ then $pi = 2$ else $pi = 1$. Of course, this is only one of the variety of ways to define the secret password. Here, the fact that the binary representation has $8k$ bits guarantees that we can always have two distinct strings of bits with length k and therefore surely have choice of strings with the required properties. Note that $e_1 d_1 = d_1 e_1 = e_2 d_2 = d_2 e_2 = 1$ $(= identitytransformation)$ and the above rule does not take care of avoiding this. In the program this is fixed by using opposite values to those in the sequence $b'_1, b'_2, \ldots, b'_k$ when such a case is encountered.

The package with the application for Linux platforms can be obtained at *http : //ii.pmf.ukim.edu.mk/ftp/crypto*. It contains the following files:

ytalk - the binary executable

kluc.dat - the public key file.

The encryption option can be specified as an option at the command line in the following format:

ytalk -e password username

For example,

ytalk -e vu}t67%Aw student

means that when chatting with the user with username **student** the sequence of leaders $l_1, l_2, \ldots, l_9$ in the enciphering process is v,u,},t,6,7,%,A,w, and the encryption function $T_9 = d_2 e_1 e_2 e_2 d_1 e_2 e_1 d_2 d_2$.

References

[1] Dénes, J., Keedwell, A.D., Latin Squares and their Applications, English Univer. Press Ltd., 1974

[2] Kościelny, C., A method of constructing quasigroup-based stream-ciphers. Appl. Math. and Comp. Sci. **6** (1996) 109–121

[3] Markovski, S., Gligoroski, D., Andova, S., Using quasigroups for one-one secure encoding. Proc. VIII Conf. Logic and Computer Science "LIRA '97", Novi Sad, (1997) 157–162

[4] Markovski, S., Gligoroski, D., Bakeva, V., Quasigroup String Processing: Part 1 (to appear in Proceedings of Mac. Akad. Scien. and Arts)

[5] Markovski, S., Kusakatov, V., Quasigroup String Processing: Part 2 (preprint)

[6] Menezes, A., van Oorchot, P., Vanstone, S., Handbook of Applied Cryptography, CRC Press, 1997

[7] Stinson D.R., Cryptography - Theory and Practice, CRC Press, 1995

[8] Tanenbaum A.S., Computer Networks, Prentice-Hall, 1996

[9] http://www.pgpi.org/