

**SIXTH INTERNATIONAL CONFERENCE
on Discrete Mathematics and Applications
31.08.2001 - 02.09.2001, Bansko, Bulgaria**

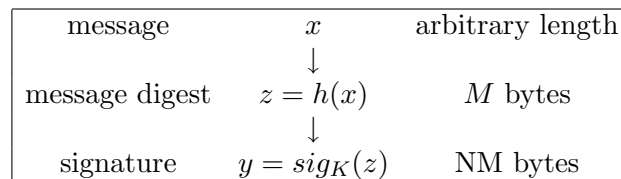
QUASIGROUPS AND HASH FUNCTIONS

SMILE MARKOVSKI, DANILO GLIGOROSKI, VERICA BAKEVA

Abstract. In this paper we consider two quasigroup transformations $QM1 : A^{2m} \rightarrow A^{2m}$ and $QM2 : A^m \rightarrow A^{2m}$, where A is the carrier of a quasigroup. Based on these transformations we show that different kinds of hash functions can be designed with suitable security.

1. PRELIMINARIES

Hash functions are a special kind of (public) functions which are used in cryptography for different cryptographic purposes, like signature schemes, message authentication codes, etc. From an input message of arbitrary length, they produce an output *message digest* of a specified size (M bytes). The obtained message digest can be used for signing large documents, for checking the validity of packages sent over insecure networks and so on.



The hash functions can be classified into two classes ([7]): the class of a weakly collision-free and the class of strongly collision-free hash functions. A hash function f is weakly collision-free if, for a given message x , it is computationally infeasible to find a message $y \neq x$ such that $f(y) = f(x)$, and it is strongly collision-free if it is computationally infeasible to find messages y and x such that $y \neq x$ and $f(y) = f(x)$. Clearly, each strongly collision-free hash function is weakly collision-free, too. There are several methods for construction

of hash functions. Here, we propose some ideas how to use a quasigroup in order to construct hash functions.

In what follows we will work with the ASCII alphabet $A = \{0, 1, \dots, 255\}$ and we denote by $A^+ = \{x_1x_2\dots x_k \mid x_i \in A, k \geq 1\}$ the set of all finite strings over A , considered as messages. (Further on, we will consider the elements of A^k as strings of length k). We denote by $M \geq 1$, a fixed positive integer such that M is the length of message digests in bytes (for application purposes usually $M \geq 64$). Then a hash function is a mapping $f : A^+ \rightarrow A^M$ with suitable properties in order to prevent various forgeries.

The quasigroup enciphering method is defined as follows. Recall that a quasigroup $(Q, *)$ is a groupoid having Caley scheme main body that is a Latin square. Let $*$ be a quasigroup operation on the set A . For a fixed letter $l \in A$, called leader, we define a transformation $d_l : A^+ \rightarrow A^+$ as follows:

$$d_l(x_1x_2\dots x_n) = y_1y_2\dots y_n \iff \begin{cases} y_1 &= l * x_1 \\ y_i &= x_{i-1} * x_i, \quad i = 2, \dots, n \end{cases}$$

where $x_i, y_i \in A, n \geq 1$. Using leaders l_i and the transformations $d_{l_i}, i = 1, 2, \dots, k$, we define a transformation $D_{l_1l_2\dots l_k} : A^+ \rightarrow A^+$ as a composition of the transformations d_{l_i} , i.e.

$$D_{l_1l_2\dots l_k} = d_{l_1} \circ d_{l_2} \circ \dots \circ d_{l_k}.$$

Proposition 1 [4] *For fixed leaders l_1, l_2, \dots, l_k , the mapping $D_{l_1l_2\dots l_k}$ is a bijection from A^+ onto A^+ .*

Proposition 2 [5] *Let $\alpha = a_1a_2\dots a_n \in A^+$ and $\beta = D_{l_1l_2\dots l_k}(\alpha)$, for some fixed leaders l_1, l_2, \dots, l_k . Then the distribution of substrings of β of length t , for each fixed t , converge to uniform distribution for enough large k and n .*

As consequence of *Proposition 1* and *Proposition 2* we have that the transformation $D_{l_1l_2\dots l_k}$ can be used for cryptography purposes.

Given $b_i \in A$ and $a \in A$, it follows that the equation $d_{x_0}(x_1\dots x_{i-1}ax_{i+1}\dots x_n) = b_1\dots b_n$ has unique solutions $x_0, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n \in A$. Namely, we have

$$b_1 = x_0 * x_1, \dots, b_{i-1} = x_{i-2} * x_{i-1}, \quad b_i = x_{i-1} * a,$$

$$b_{i+1} = a * x_{i+1}, \quad b_{i+2} = x_{i+1} * x_{i+2}, \dots, b_n = x_{n-1} * x_n.$$

That implies at first stage unique solutions for x_{i-1} and x_{i+1} , after that unique solutions for x_{i-2} and x_{i+2} , and so on. Thus, we have the following:

Proposition 3 *Let $d_{x_0}(x_1\dots x_n) = b_1\dots b_n$, where b_1, \dots, b_n are known, and x_0, x_1, \dots, x_n are unknown letters of A . If we choose a value (i.e. a fixed*

letter of A) for one of the unknown letters x_i , the values of the other ones are uniquely determined.

2. ON THE SECURITY OF QUASIGROUP TRANSFORMATIONS

Quasigroup method 1 (QM1)

Let $\text{bin} : A \rightarrow \mathbb{N}$ be a mapping from the set A to \mathbb{N} defined by $\text{bin}(x) = k$, where k is the sum of the digits in the binary presentation of the element $x \in A$. Further on, we will use the following notation \bullet , defined for each $n \geq 1$ as follows:

$$(1) \quad x_1 \bullet x_2 \bullet \cdots \bullet x_n = (\dots ((x_n * x_{n-1}) * x_{n-2}) * \cdots * x_2) * x_1$$

Let $c_1 c_2 \dots c_{2m}$ be a given string with length $2m$. We take an even number $2m$ only for technical reasons. We apply the quasigroup transformation $D_{l_0 l_1 \dots l_m}$ on that string where the leaders l_0, l_1, \dots, l_m and the intermediate values c_j^i are defined by the following way. For each $i = 0, 1, 2, \dots, m$, let

$$d_i = \left(\sum_{j=1}^{2m} \text{bin}(c_j^i) \right) (\text{mod } m) + m + 1,$$

$$l_i = c_1 \bullet c_2 \bullet \cdots \bullet c_{d_i},$$

where $c_j^0 = c_j$, for $j = 1, 2, \dots, 2m$, $c_1^i c_2^i \dots c_{2m}^i = d_{l_{i-1}}(c_1^{i-1} c_2^{i-1} \dots c_{2m}^{i-1})$, for $i = 1, \dots, m$. It is obvious that $m < d_i \leq 2m$. Let $D_{l_0 l_1 \dots l_m}(c_1 \dots c_{2m}) = g_1 \dots g_{2m}$. Then we have that the Table 1 is fulfilled.

l_0	c_1	c_2	\dots	c_{2m}
l_1	c_1^1	c_2^1	\dots	c_{2m}^1
l_2	c_1^2	c_2^2	\dots	c_{2m}^2
\vdots	\vdots	\vdots	\ddots	\vdots
l_m	c_1^m	c_2^m	\dots	c_{2m}^m
	g_1	g_2	\dots	g_{2m}

Table 1

According to the definition of the quasigroup transformation $D_{l_0 l_1 \dots l_m}$, we can note that the g_i depends on c_1, c_2, \dots, c_{2m} , i.e. g_i is a function of c_1, c_2, \dots, c_{2m} , for each $i = 1, \dots, 2m$. Note that each of the leaders l_i depends on c_1, c_2, \dots, c_{2m} as well.

Let a sequence $g_1 \dots g_{2m} \in A^{2m}$ be given. We will show that it is computationally infeasible to find a sequence $x_1 \dots x_{2m} \in A^{2m}$ such that $x_1 \dots x_{2m} \neq c_1 \dots c_{2m}$ and

$$(2) \quad D_{l'_0 l'_1 \dots l'_m}(x_1 x_2 \dots x_{2m}) = g_1 \dots g_{2m}.$$

In other words, we try to find $x_i^j \in A$ such that a table of the following kind is fulfilled

l'_0	x_1	x_2	\dots	x_{2m}
l'_1	x_1^1	x_2^1	\dots	x_{2m}^1
\vdots	\vdots	\vdots	\ddots	\vdots
l'_m	x_1^m	x_2^m	\dots	x_{2m}^m
	g_1	g_2	\dots	g_{2m}

Table 2

where $x_i^j = x_{i-1}^{j-1} * x_i^{j-1}$ ($x_0^0 = l'_0$, $x_i^0 = x_i$, $x_i^{m+1} = g_i$, $x_0^j = l'_j$), for $i = 1, 2, \dots, 2m$ and $j = 1, 2, \dots, m$ and the leaders l'_j for $j = 0, 1, \dots, m$ are determined as previous (as in (3)).

Since $g_1 \dots g_{2m}$ are known (given), by *Proposition 3*, we have that if anyone of the unknown $l'_m, x_1^m, x_2^m, \dots, x_{2m}^m$ obtains some value then the values of the other unknowns are uniquely determined. Thus, it is enough to give a value of l'_m , and it can be done in 256 different ways. The procedure can continue upward for the next rows, we can choose a value of l'_{m-1} (and there are 256 different ways for that), then the values of $x_1^{m-1}, x_2^{m-1}, \dots, x_{2m}^{m-1}$ are uniquely determined, and so on until we compute the elements x_1, x_2, \dots, x_{2m} . After that we have to check if the elements l'_i are well chosen, i.e. to check if the following equalities are true:

$$(3) \quad l'_i = x_1 \bullet x_2 \bullet \dots \bullet x_{d_i}, \quad \text{where} \quad d_i = \sum_{j=1}^{2m} \text{bin}(x_j^i) \pmod{m} + m + 1$$

for $i = 0, 1, \dots, 2m$ and x_j^i obtained from Table 2.

Here (3) is a resolvent system for checking if leaders are well chosen. In the worst case, we have to make 256^{m+1} checks for surely finding x_1, x_2, \dots, x_{2m} such that (2) is satisfied. We know that $x_1 = c_1, \dots, x_{2m} = c_{2m}$ is one solution, but existence of some other solutions depends on \bullet , i.e depends on the quasigroup operation $*$.

The conditions (2) and (3) are equivalent and they can be considered in another equivalent way. Since x_k^j depends only on x_1, \dots, x_{2m} , we can replace x_k^j in (3) by expressions in which x_1, \dots, x_{2m} appear. Then we get a system (3') of $4m + 2$ equations with $2m$ unknowns. If one can solve (3') then the solution will satisfy (3). Trivially, (3') has a solution $(c_1, c_2, \dots, c_{2m})$, but we need to find another one. Conversely, if x_1, \dots, x_{2m} satisfy (3), they will satisfy (3'), too. The system of equations (3') is complex one, and we give a simple illustration in the next example.

Example Let $g_1 g_2 g_3 g_4$ be given. For $2m = 4$ we have a system (3') with 10 equations and 4 unknown variables x_1, x_2, x_3, x_4 :

$$\left\{ \begin{array}{l} l_2 * (l_1 * (l_0 * x_1)) = g_1 \\ (l_1 * (l_0 * x_1)) * ((l_0 * x_1) * (x_1 * x_2)) = g_2 \\ ((l_0 * x_1) * (x_1 * x_2)) * ((x_1 * x_2) * (x_2 * x_3)) = g_3 \\ ((x_1 * x_2) * (x_2 * x_3)) * ((x_2 * x_3) * (x_3 * x_4)) = g_4 \\ d_0 = (\text{bin}(x_1) + \text{bin}(x_2) + \text{bin}(x_3) + \text{bin}(x_4)) \pmod{2} + 3 \\ d_1 = (\text{bin}(l_1 * x_1) + \text{bin}(x_1 * x_2) + \text{bin}(x_2 * x_3) + \text{bin}(x_3 * x_4)) \pmod{2} + 3 \\ d_2 = (\text{bin}(l_1 * (l_0 * x_1)) + \text{bin}((l_0 * x_1) * (x_1 * x_2)) + \text{bin}((x_1 * x_2) * (x_2 * x_3)) + \\ \quad + \text{bin}((x_2 * x_3) * (x_3 * x_4))) \pmod{2} + 3 \\ l_i = x_1 \bullet \dots \bullet x_{d_i}, \quad i = 0, 1, 2 \end{array} \right. .$$

Considering the security of a hash function that can be defined by this method, we state the following hypotheses.

Hypotheses 1 *If $2m \geq 64$, then the solution of the system (3') is computationally infeasible.*

Now, we can state that the security of quasigroup transformation is based on the computational infeasibility to find a solution of systems of quasigroup equations in a given quasigroup, in which the binary structure of the characters is involved. Such a system can have unique solution, more than one solution or do not have any solution at all.

Quasigroup method 2 (QM2)

Let suppose that $a_1 a_2 \dots a_m$ be a given string and $a'_1 a'_2 \dots a'_m = d_l(a_1 a_2 \dots a_m)$, where $l = a_1 \oplus a_2 \oplus \dots \oplus a_m$. Here \oplus denotes the addition modulo 256. We consider the string

$$z_1 z_2 \dots z_{2m} = a_1 a'_1 a_2 a'_2 \dots a_m a'_m$$

and define the sequence $g_1 g_2 \dots g_{2m}$ on the following way:

$$g_i = z_i \oplus (z_{i-1} * z_i), \quad i = 1, 2, \dots, 2m, \quad \text{where } z_0 = l.$$

We can denote the last operations on a short way as

$$g_1 g_2 \dots g_{2m} = z_1 z_2 \dots z_{2m} \oplus d_l(z_1 z_2 \dots z_{2m}).$$

Note that $z_{2i+1} = a_i$ and $z_{2i} = a'_i = a_{i-1} * a_i$ for $i = 1, \dots, m$.

We want to find another sequence $y_1 y_2 \dots y_m (\neq a_1 a_2 \dots a_m)$ such that

$$g_1 g_2 \dots g_{2m} = y_1 y'_1 y_2 y'_2 \dots y_m y'_m \oplus d_{l_1}(y_1 y'_1 y_2 y'_2 \dots y_m y'_m),$$

where $l_1 = y_1 \oplus y_2 \oplus \dots \oplus y_m$. In order to find such a sequence $y_1 y_2 \dots y_m$ we have to solve a system of the following kind:

$$\begin{aligned}
 (4) \quad & l_1 = y_1 \oplus y_2 \oplus \dots \oplus y_m \\
 & y_1 \oplus (l_1 * y_1) = g_1 \\
 & (l_1 * y_1) \oplus (y_1 * (l_1 * y_1)) = g_2 \\
 & y_2 \oplus ((l_1 * y_1) * y_2) = g_3 \\
 & (y_1 * y_2) \oplus (y_2 * (y_1 * y_2)) = g_4 \\
 & \vdots \\
 & (y_{m-3} * y_{m-2}) \oplus (y_{m-2} * (y_{m-3} * y_{m-2})) = g_{2m-4} \\
 & y_{m-1} \oplus ((y_{m-3} * y_{m-2}) * y_{m-1}) = g_{2m-3} \\
 & (y_{m-2} * y_{m-1}) \oplus (y_{m-1} * (y_{m-2} * y_{m-1})) = g_{2m-2} \\
 & y_m \oplus ((y_{m-2} * y_{m-1}) * y_m) = g_{2m-1} \\
 & (y_{m-1} * y_m) \oplus (y_m * (y_{m-1} * y_m)) = g_{2m}
 \end{aligned}$$

We can try to find a solution of the previous system starting from the last equation. From all 256^2 possible pairs $(y_{m-1}, y_m) \in A^2$, we have to find all of them which satisfy the last equation of the system. Given any solution (y_{m-1}, y_m) of the last equation, from the equation before the last, we can find y_{m-2} on the unique way. The next equation upward is a control equation for checking if y_{m-2}, y_{m-1}, y_m are well chosen. From the next equation y_{m-3} is uniquely determined, and the equation next upward of it, is a control equation again and so on. Actually, at first we can consider the subsystem:

$$\begin{aligned}
 (5) \quad & y_i \oplus ((y_{i-2} * y_{i-1}) * y_i) = g_{2i-1}, \quad i = 2, 3, \dots, m \\
 & (y_{m-1} * y_m) \oplus (y_m * (y_{m-1} * y_m)) = g_{2m}
 \end{aligned}$$

where $y_0 = l_1$ is taken to be the leader in the quasigroup transformation. Choosing the solution (y_{m-1}, y_m) of the last equation, the other unknowns y_i , $i = m-2, m-3, \dots, 1, 0$ are uniquely determined, but we have to check if they satisfy the following equations:

$$\begin{aligned}
 (6) \quad & y_0 = y_1 \oplus y_2 \oplus \dots \oplus y_m \\
 & y_1 \oplus (y_0 * y_1) = g_1 \\
 & (y_{i-1} * y_i) \oplus (y_i * (y_{i-1} * y_i)) = g_{2i}, \quad i = 1, 2, \dots, m
 \end{aligned}$$

We note that for any given $g_1, g_2, \dots, g_{2m} \in A$ the system (5) has exactly 256 solutions. Namely, if we denote $a = (y_{m-1} * y_m)$ and $b = y_m * (y_{m-1} * y_m)$, the last equation in the system (5) can be rewritten as $a \oplus b = g_{2m}$ and it has 256 solutions (for each of 256 different values of a , the values of b are uniquely

determined). For obtained values of $a = y_{m-1} * y_m$ and $b = y_m * a$, we can compute y_{m-1} and y_m from the quasigroup $(A, *)$ on a unique way. After that y_i for $i = m-2, \dots, 2, 1$ are uniquely determined. This means that the system (5) has exactly 256 solutions. But, each of them has to satisfy $m+2$ equations in (6). We can choose m enough large such that the possibility to find another solution of the system (4) different than $a_1 a_2 \dots a_m$, is enough small. In other words, we state the following

Hypotheses 2 If m is large enough, then the system of equations (4) has no solutions different than the given one.

3. CONCLUSION

We presented two quasigroup methods for transformation of strings of length $2m$ and of length m , respectively, into strings of length $2m$. The security of these transformations, based on Hypotheses 1 and Hypotheses 2, was considered too. These transformations can be applied for designing suitable hash functions in many different ways, and here we mention a few of them.

Let $a_1 a_2 \dots a_r \in A^r$, $r \geq 1$. If we apply QM1 and $r \not\equiv 0 \pmod{2m}$, we concatenate to $a_1 a_2 \dots a_r$ a string $10 \dots 0$ such that the new string $a_1 a_2 \dots a_r 10 \dots 0 = a_1 a_2 \dots a_{2mn}$ has the length $2mn$ for some smallest possible $n \geq 1$. Then we separate the string $a_1 a_2 \dots a_{2mn}$ on n blocks (substrings)

$$B_i = a_{2m(i-1)+1} a_{2m(i-1)+2} \dots a_{2mi}, \quad i = 1, 2, \dots, n,$$

each of length $2m$, and we apply QM1 to everyone of the blocks B_i . Let

$$\text{QM1}(B_i) = g_1^i g_2^i \dots g_{2m}^i.$$

Now, a hash function H_1 can be defined by

$$H_1(a_1 \dots a_r) = h_1 \dots h_{2m}$$

where

$$h_i = \bigoplus_{j=1}^n g_i^j, \quad i = 1, 2, \dots, 2m.$$

H_1 produces message digests of length $M = 2m$ and it is a strongly collision-free hash function.

The QM2 can be used in the same way as QM1, but we present another design. Given message $a_1 \dots a_r$, if $r < m$ it can be enlarged to a message $a_1 a_2 \dots a_r 10 \dots 0 = a_1 a_2 \dots a_m$. So, let $r \geq m$. Now, we apply QM2 as

follows. Let $j = r - m$ and

$$QM2(a_1 \dots a_m) = g_1^1 \dots g_{2m}^1$$

$$QM2(g_{m+2}^1 \dots g_{2m}^1 a_{m+1}) = g_1^2 \dots g_{2m}^2$$

$$QM2(g_{m+2}^2 \dots g_{2m}^2 a_{m+2}) = g_1^3 \dots g_{2m}^3$$

$$\vdots$$

$$QM2(g_{m+2}^j \dots g_{2m}^j a_r) = g_1^{j+1} \dots g_{2m}^{j+1}$$

Then we define a hash function H_2 by

$$H_2(a_1 \dots a_r) = g_1^{j+1} \dots g_{2m}^{j+1},$$

which produces message digests of length $2m$.

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, Contributions, Sec. Math. Tech. Sci., MANU, XX,1-2 (1999) 13–28
- [5] Markovski, S., Kusakatov, V.: Quasigroup String Processing: Part 2, Contributions, Sec. Math. Tech. Sci., MANU (in print)
- [6] Markovski, S., Gligoroski, D., Stojčevska, B.: Secure two-way on-line communication by using Quasigroup Enciphering with almost public key, Novi Sad J. Math. Vol. 30, No.2, 2000, 43–49
- [7] Stinson, D. R.: Cryptography – Theory and Practice, CRC Press, New York, 1995

The Faculty of the Natural Sciences and Mathematics,

Institute of Informatics,

P.O.Box 162, Skopje,

Republic of Macedonia

{smile, daniilo, verica}@pmf.ukim.edu.mk

<http://ii.pmf.ukim.edu.mk/crypto/>