Tatra Mt. Math. Publ. **29** (2004), 173–181

TATRA MOUNTAINS
Mathematical Publications

CRYPTANALYSIS OF ONE HASH FUNCTION BASED ON QUASIGROUP

MILAN VOJVODA

ABSTRACT. We discuss to what extent hash functions proposed in [J. Dvorský, E. Ochodková, V. Snášel: *Hash Function Based on Quasigroups*, Proc. of Mikulášská kryptobesídka, Praha, pp. 27–36, (2001) (in Czech)] [J. Dvorský, E. Ochodková, V. Snášel: *Hash Functions Based on Large Quasigroups, Proc. of Velikonoční kryptologie*, Brno, pp. 1–8, (2002)] and defined on quasigroups, are resistant to several kinds of attacks. We present an attack against these hash functions for some special quasigroups. In details we analyze the quasigroup of modular subtraction, which is mainly in question. Some further practical considerations will be given, too.

1. Introduction

It can simply be said that hash functions compress a string over an alphabet (usually $\{0,1\}$) of arbitrary length to a string of fixed length. They have a large number of applications in computer science (optimized (fast) access to the stored data) and as well as in cryptography (integrity protection of stored/transmitted data). However, hash functions suitable for use in cryptographic applications may need to meet stronger requirements than for other applications.

According to the number of inputs one can subdivide the hash functions into two classes:

- one input (a message to be hashed). These are called Manipulation Detection Codes (MDCs), sometimes also cryptographic hash functions or just only hash functions.
- two inputs (a message to be hashed and a key). If the key is kept secret one calls them Message Authentication Codes (MACs). If the key is public one calls them Universal One-Way Hash Functions (UOWHFs).

2000 Mathematics Subject Classification: 94A60.
Keywords: hash function, quasigroup, cryptanalysis.

This work was supported by VEGA-grant 1/0161/03.

Next we give an informal basic definition of a hash function and of a collision resistant hash function.

Definition 1. [6] Let $\{0,1\}^*$ be the set of all finite words over the alphabet $\{0,1\}$. A *one-way hash function* is a function $h\colon \{0,1\}^* \to \{0,1\}^m$ satisfying the following conditions:

1. The argument x can be of arbitrary length and the result $h(x)$ has a fixed length of m bits. Nowadays $m \geq 128, \dots, 256$.
2. The hash function must be one way in the sense that given a y in the image of h, it is "hard" to find a message x such that $h(x) = y$ (preimage resistant) and given x in the domain of h and $h(x)$ it is "hard" to find a message $x' \neq x$ such that $h(x') = h(x)$ (second preimage resistant).

Definition 2. [6] A *collision resistant hash function* is a function h satisfying the following conditions:

1. The argument x can be of arbitrary length and the result $h(x)$ has a fixed length of m bits. Nowadays $m \geq 128, \dots, 256$.
2. The hash function must be one way, i.e., preimage and second preimage resistant.
3. That hash function must be collision resistant, i.e., it is "hard" to find two distinct messages x, x' such that $h(x) = h(x')$.

Clearly, both definitions given above can be extended for any alphabet Q.

The usage of quasigroups (resp. latin squares) in cryptography is not very common. In spite of that, various cryptosystems based on quasigroups appeared in past few years, e.g., see [4], where it is shown that the usage of quasigroups in the design of S-boxes might open new ways in the design of block ciphers.

Definition 3. [1] The structure $(Q, \odot)$, $Q = \{q_1, q_2, \dots, q_n\}$, $\|Q\| = n$ is called a *finite quasigroup* of order n if, when any two elements $a, b \in Q$ are given, the equations $a \odot x = b$ and $y \odot a = b$ each have exactly one solution. Thus the Caley table of a finite quasigroup of order n is a latin square, i.e., an $n \times n$ array with the property that each row and each column contains a permutation of symbols from Q.

Definition 4. [1] Let $(G, \cdot)$ and $(H, \odot)$ be two quasigroups. An ordered triple (θ, φ, ψ) of one-to-one mappings θ, φ, ψ of the set G onto the set H is called an *isotopism* of $(G, \cdot)$ upon $(H, \odot)$ if $\theta(x) \odot \varphi(y) = \psi(x \cdot y)$ for all x, y in G. The quasigroups $(G, \cdot)$ and $(H, \odot)$ are then said to be *isotopic*.

The structure of this paper is as follows. The new construction of a hash function based on a quasigroup, proposed in [2], [3] is given in Section 2. Several attacks against the studied hash function are proposed in Section 3. Security of the studied hash function when used as MAC is studied in Section 4, where several attacks are presented. Finally, conclusions are given in Section 5.

2. Construction of a new hash function

Construction 5. (A new hash function [2], [3].) *Let $(Q, \odot)$ be a finite quasigroup. Let us denote the message as a sequence of elements $(m_1, m_2, \ldots, m_k)$ from the quasigroup Q. Let Q^* be the set of all finite words over Q. The hash function $H_a : Q \times Q^* \to Q$, $a \in Q$, is fixed, is given as*

$$H_a(m_1, m_2, \ldots, m_k) = \big(\big(\ldots\big(\big(a \odot m_1\big) \odot m_2\big) \odot \ldots\big) \odot m_{k-1}\big) \odot m_k\,,$$

where $m_i \in Q$, $1 \le i \le k$.

Example 6. Let $Q = \{0, 1, 2, 3\}$ and let the operation $\odot$ defined on Q be defined by its Caley table, see Table 1.

Table 1. Caley table of the operation $\odot$ defined on Q.

$\odot$	0	1	2	3
0	0	2	1	3
1	2	3	0	1
2	1	0	3	2
3	3	1	2	0

Let $a = 2$ and let the message to be hashed be encoded as $(0, 0, 1, 3)$. Then the hash can be computed as

$$H_2(0, 0, 1, 3) = \big(\big((2 \odot 0) \odot 0\big) \odot 1\big) \odot 3 = 3\,.$$

The usage of a general quasigroup in computation requires to store its Caley table, i.e., n^2 elements. The storage requirements are outlined in Table 2. (One can also take notice of a very short hash value length. Nowadays the hash value length considered to be secure is 160–256 bits.).

Table 2. Storage requirements for the Caley table a general quasigroup

$\lVert Q\rVert$	Hash value length	Storage requirements
2^{16}	16 bits	$16.2^{16}.2^{16} = 64$ GB
2^{18}	18 bits	$18.2^{18}.2^{18} > 1$ TB

Trivially, the last column and the last row need not be stored. One can also use the known results from critical sets in latin squares. It is known that a latin square is uniquely determined by $n^2/4$ of its cells. It is also known that the minimal number of cells to be known for a unique completion of a latin square is $\lfloor(4n-8)/3\rfloor$, for $n \geq 8$ (see [5]). However, this approach will significantly slow down the speed of computation.

In order to overcome the storage requirements for the Caley table, a special quasigroup, namely the quasigroup of modular subtraction, was proposed in [2], [3] to be used. The operation $\odot$ defined on Q is then given as

$$a \odot b = a + n - b \mod n\,, \qquad n = \|Q\|\,.$$

TABLE 3. Multiplication table in the quasigroup of modular subtraction, $n = 4$

$\odot$	0	1	2	3
0	0	3	2	1
1	1	0	3	2
2	2	1	0	3
3	3	2	1	0

Usage of such an "easy to evaluate" expression for the definition of the operation $\odot$ on quasigroup allows us to use quasigroups with a very large number of elements. Moreover, the isotopism of quasigroups gives us the power to use a large number of isotopic quasigroups where the computation of a hash value will be done almost only using the mentioned "easy to evaluate" expression. As it will be shown later, it has also a severe impact on the security of the studied hash function.

3. Attacks against the hash function

Let $a \in Q$ be the known parameter of the hash function and $(m_1, m_2, \dots m_k)$, $m_i \in Q$, $1 \leq i \leq k$, be a message to be hashed. Let us denote the hash value as

$$H_a(m_1, m_2, \dots, m_k) = \big(\dots\big((a \odot m_1) \odot m_2\big) \odot \dots\big) \odot m_k = d\,.$$

Due to the simple construction of the hash function one can (in some cases easily) create false messages that hash to the same value. The false message can be constructed from the original message by adding a prefix and/or suffix, changing some parts somewhere in the middle of the message, or it can be just a totally new message not based on the original message.

The false message created from the original one by adding a prefix can be written as

$$(p_1, p_2, \dots, p_l, m_1, m_2, \dots m_k)\,, \qquad p_i \in Q\,,\ 1 \leq i \leq l\,.$$

Hence it must hold that $(\dots((a \odot p_1) \odot p_2) \odot \dots) \odot p_l = a$.

The false message created from the original one by adding a suffix can be written as

$$(m_1, m_2, \dots m_k, s_1, s_2, \dots, s_t), \qquad s_i \in Q\,,\ 1 \le i \le t\,.$$

Hence it must hold that $(\dots((d \odot s_1) \odot s_2) \odot \dots) \odot s_t = d$.

Remark 7. Only the last element of the last added, resp. changed part of the message has to be chosen in a proper way. All the other elements can be chosen arbitrary, i.e., they can represent meaningful data.

Altering a part of the message or creating a new false message is similar to previous examples of false messages.

Theorem 8. *The complexity of producing a false message (using the construction given above in this section) is at most of order* $\|Q\|$.

Remark 9. The proof of resistance of the studied hash function against attacks presented in [3] is misleading. Assume that a hash value of some message was given to an attacker. His/her task is not finding the original message from the hash value! For the attacker is sufficient to find any message that hash to the specified hash value. Moreover in practice one usually looks for a different message from the original one.

Remark 10. While thinking about altering some parts of the original message one may ask how many elements may/must be changed in order to get the same hash value as the original message. It can be easily seen that changing a single element leads always to a hash value different from the hash value of the original message. Further it can be seen that changing more than one element in the original message always allows to reach the same hash value as the original message. Note that only the last element has to be chosen properly (see below).

The question is how to find the last element of the last added/changed/created part of the false message. In other words, if we want to produce a false message by adding prefix to the original message, how to find p_l such that the following will hold $(\dots((a \odot p_1) \odot p_2) \odot \dots) \odot p_l = a$?

For general small (i.e., with storable Caley table) quasigroups it is possible to make an exhaustive search, i.e., a brute force attack (see Table 2). It is widely accepted that problems of complexity up to 2^{64} are nowadays solvable by an exhaustive search. Thus the table implementation of the studied hash function is not secure.

In order to overcome the storage requirements for the Caley table, a special quasigroup, namely the quasigroup of modular subtraction, was proposed in [2], and [3].

Lemma 11. *Quasigroup of modular subtraction contains a right unit, namely 0.*

Theorem 12. *Let the quasigroup of modular subtraction be used. Then to construct a false message one can insert an arbitrary number of 0s anywhere into the original message.*

The last theorem shows a trivial construction of false messages based on the insertion of a right unit. However, one can do much more.

Let us try to create a totally new message $(x_1, x_2, \dots, x_v)$, $x_i \in Q$, $1 \le i \le v$, that will hash to the value d. The elements $x_1, x_2, \dots, x_{v-1}$ can be chosen arbitrary. Let $d' = \big(\dots\big((a \odot x_1) \odot x_2\big)\dots\big) \odot x_{v-1}$. It remains to find such a x_v that $d' \odot x_v = d$,

$$x_v = d' + n - d \mod n\,.$$

Consequence 13. *Hash function H_a with the quasigroup of modular subtraction is neither collision resistant, nor second preimage resistant, nor preimage resistant.*

The attack may become much more difficult when there is a quasigroup isotopic to the quasigroup of modular subtraction used for the hash function.

Example 14. Let $(Q, \odot)$, $\|Q\| = 4$, be the quasigroup of modular subtraction with the Caley table given in Table 3. Let $\theta = [1,2,3,0]$, $\varphi = [3,2,1,0]$ and $\psi = [2,0,3,1]$. The Caley table of the quasigroup $(Q, \cdot)$ isotopic to the $(Q, \odot)$ is shown in Table 4.

Table 4. Multiplication table of the quasigroup $(Q, \cdot)$.

$\cdot$	0	1	2	3
0	0	2	1	3
1	2	1	3	0
2	1	3	0	2
3	3	0	2	1

The quasigroup operation in $(Q, \cdot)$ (that is isotopic to quasigroup of modular subtraction) may be written also as

$$a.b = \psi^{-1}\big(\theta(a) + n - \varphi(b) \mod n\big)\,, \qquad n = \|Q\|\,.$$

In the previously described attacks the attacker was forced to solve the equation $a \odot b = d$ in a given quasigroup, where a and d were known, and b was unknown. Formally the solution can be written as $b = a\backslash d$, where $\backslash$ is the right inverse of $\odot$ (note that the following holds: $x\backslash(x \odot y) = y$, and $x \odot (x\backslash y) = y$). For a quasigroup of modular subtraction one can write $b = a\backslash d = a \odot d$.

The security of the studied hash function severely depends on the difficulty of inverting the one-to-one mappings φ and ψ^{-1}.

There might be also another security problem. Let $(Q,\cdot)$ and $(Q,\odot)$ be two isotopic quasigroups, i.e., there exist one-to-one mappings θ,φ,ψ of the set Q onto the set Q, such that $\theta(x)\odot\varphi(y)=\psi(x\cdot y)$ for all x,y in Q. However, there might exist one-to-one mappings θ',φ',ψ' of the set Q onto the set Q, such that $\theta'(x)*\varphi'(y)=\psi'(x\cdot y)$ for all x,y in Q. For example another triplet of mappings (θ,φ,ψ) that define isotopism between the quasigroups used in Example 14 is $\theta=[1,2,3,0]$, $\varphi=[2,1,0,3]$ and $\psi=[3,1,0,2]$. It might happen that even though the mappings φ and ψ^{-1} are "hard to invert", the mappings φ' and ${\psi'}^{-1}$ are "easy to invert". However, finding such mappings θ',φ',ψ' may be difficult.

We performed an exhaustive search experiments (for $\|Q\|=3,4,5$, and 6) where we studied the number of one-to-one mappings θ,φ,ψ that define isotopism between the quasigroup of modular subtraction $(Q,\odot)$ and any quasigroup $(Q,\cdot)$ isotopic to this quasigroup of modular subtraction. In all the experiments the number of one-to-one mappings θ,φ,ψ was $2\|Q\|^2$.

4. Keyed hash function

Assume a in the hash function H_a is a secret key. H_a is then MAC. Assume that the quasigroup of modular subtraction was used. In the following we will show how to create false messages, that will hash to the same value.

Let $(m_1,m_2,\dots m_k)$, $m_i\in Q$, $1\le i\le k$, be a message to be hashed. Let us denote the hash value as

$$H_a(m_1,m_2,\dots,m_k)=\big(\dots\big((a\odot m_1)\odot m_2\big)\odot\dots\big)\odot m_k=d\,.$$

We can add a suffix and construct such a message $(m_1,m_2,\dots,m_k,s_1,s_2,\dots,s_u)$, $s_i\in Q$, $1\le i\le u$, that will hash to the same result as the original message. Again the elements $s_1,s_2,\dots,s_{u-1}$ can be arbitrary (i.e., they can represent meaningful data), only s_u has to be calculated in a proper way. For the quasigroup of modular subtraction it is easy to do (see Section 3) it.

It is possible to create the following false messages that will hash to the same value as the original message: a totally new message, add suffix to the original message, or change some parts of the message. Adding only a prefix to the original message seems to be impossible due to the secret key a. However, it is possible to add both a prefix and a suffix, or add a prefix and change some part of the original message.

COROLLARY 15. *A hash function H_a with the quasigroup of modular subtraction when used as MAC, with the secret key a, is neither collision resistant, nor second preimage resistant, nor preimage resistant.*

5. Conclusions

There were some attacks against the hash function, proposed in [2], [3], shown in this paper. Attacks were studied in a setting when a general (storable, i.e., small) quasigroup was used and also when a special (large) quasigroup, namely the quasigroup of modular subtraction, was used. The security of the construction of a hash functions was studied in the MDC and also in the MAC scenario. In all the cases it was possible to create false messages as it can be seen in Sections 3, and 4.

In order to make such a hash function useful in cryptology, a very special quasigroup $(Q, \odot)$ has to be found. The multiplication in such a quasigroup, $a \odot b = f(a,b)$, $a, b \in Q$, should be given by some "easy to evaluate" expression $f(a,b)$. Thus a "large" quasigroup could be used (without storing its multiplication table). Moreover, given a, resp. b and $f(a,b)$, it must be "difficult" (computationally infeasible) to find b, resp. a. One of the ways how to achieve this is to use isotopic quasigroup to the quasigroup of modular subtraction, as it was proposed in [2], [3]. The security of the studied hash function when such a quasigroup is used, is a topic for further research.

Acknowledgement. The author gratefully acknowledges many helpful suggestions of Professor Otokar Grošek during preparation of the paper.

REFERENCES

[1] DÉNES, J.—KEEDWELL, A. D.: *Latin Squares and their Applications*, Academic Press, New York, 1974.

[2] DVORSKÝ, J.—OCHODKOVÁ, E.—SNÁŠEL, V.: *Hash function based on quasigroups* (*"Hashovací funkce založená na kvazigrupách"*), in: Proc. of Mikulášská kryptobesídka, Praha, 2001, pp. 27–36, 2001. (In Czech)

[3] DVORSKÝ, J.—OCHODKOVÁ, E.—SNÁŠEL, V.: *Hash functions based on large quasigroups*, in: Proc. of Velikonoční kryptologie, Brno, 2002, pp. 1–8.

[4] GROŠEK, O.—SATKO, L.—NEMOGA, K.: *Ideal difference tables from an algebraic point of view,* Cryptology and Information Security, in: Proc. of VI RECSI, Teneriffe, Spain, 2000 ammendment to CRIPTOLOGÍA y SEGURIDAD de la INFORMACIÓN, (P. Caballero-Gil and C. Hernández-Goya, eds.), RA-MA, Madrid, 2000, pp. 453–454.

[5] HORÁK, P.—ALDRED, R. E. L.—FLEISCHNER, H.: *Completing latin squares: critical sets*, J. of Combin. Des. **10** (2002), 419–432.

[6] PRENEEL, B.: *The state of hash functions,* Cryptology and Information Security, in: Proceedings of VI RECSI, Teneriffe, Spain, September 2000 (P. Caballero-Gil and C. Hernández-Goya, eds.), RA-MA, Madrid, 2000, pp. 3–27.

[7] VOJVODA, M.: *On one hash function based on quasigroup*, in: Proc. of Mikulášská kryptobesídka, Praha, ecom-monitor.com, 2003, pp. 23–28.

Received September 24, 2003

Department of Mathematics
Faculty of Electrical Engineering
and Information Technology
Slovak University of Technology
Ilkovičova 3
SK–812 19 Bratislava
SLOVAKIA

E-mail: vojvoda@kmat.elf.stuba.sk