ON INFINITE CLASS OF STRONGLY COLLISION RESISTANT HASH FUNCTIONS "EDON-F" WITH VARIABLE LENGTH OF OUTPUT

DANILO GLIGOROSKI, SMILE MARKOVSKI AND VERICA BAKEVA

"Ss Cyril and Methodius" University,
Faculty of Sciences and Mathematics,
Institute of Informatics, p.f. 162,
Skopje, Republic of Macedonia,
{verica,danilo,smile}@ii.edu.mk

ABSTRACT

Two infinite classes of strongly collision free hash functions "Edon-C" and "Edon-R" are defined in [GMB 2003]. Here we propose one more hash function of similar 'Edon' type called "Edon-F". This hash function is based on the theory of quasigroups and the cryptographic properties of quasigroup string processing, i.e. by similar idea as the others 'Edon' type of hush functions are designed. The "Edon-F" hash function is designed in such a way to be much faster than "Edon-C" and "Edon-R", i.e. it is with linear complexity. The price for that is paid on the security level, in the sense that the length of the output message (the message digest) should be enough large.

Key words: quasigroup, one-way function, hash function
AMS Mathematics Subject Classification (2000): 94A60, 20N05

1. PRELIMINARIES

Given sets M and N, a function $f : M \longrightarrow N$ is said to be a one-way function if it is easy to compute the image $f(m)$ of each $m \in M$ (i.e. there is an algorithm of polynomial complexity to compute $f(m)$), and it is computationally infeasible to find an origin m of any given $n \in N$ (i.e. for computing the origin of any $n \in N$ there are only algorithms of exponential complexity). Let A be a finite set and let A^+ denote the set of all finite strings over A. The elements of A^+ will be denoted by $a_1a_2...a_n$ where $a_i \in A$. By A^m is denoted the set of all elements of A^+ with length m. A hash function is said to be any one-way function from the set A^+ into A^m where m is some fixed positive integer. If we take the set A to be the set of ASCII code then $|A| = 256$ and any ASCII file can be considered as an element of A^+. If $h : A^+ \longrightarrow A^m$ is a hash function then the image $h(x)$ is said to be the message digest of the message x. The integer m is called digest size of the hash function h.

A hash function h is useful for cryptographic purposes if it satisfies some additional properties. If x and y are two messages that differ at least in one bit then $h(x)$ and $h(y)$ should differ in about a half of their bits. We say that a hash function h is weakly collision-free if for given message x it is computationally infeasible to find another message y ($\neq x$) such that $h(x) = h(y)$; h is said to be strongly collision-free if it is computationally infeasible to find two different messages x and y such that $h(x) = h(y)$. Any strongly collision-free hash function is weakly collision too.

A quasigroup $(Q, *)$ is a groupoid satisfying the law

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

This implies the cancellation laws $x * y = x * z \Rightarrow y = z$, $y * x = z * x \Rightarrow y = z$ and the equations $a * x = b$, $y * a = b$ have unique solutions x, y for each a, $b \in Q$. In the quasigroup theory there are not known algorithms for solving some trivial quasigroup equations, like $(x * a) * (b * x) = c$, where x is an unknown, or system of quasigroup equations. So, the way for solving an equation or system of equations over a given quasigroup $(Q, *)$ is by checking all of the possibilities. We will use this fact in proving the resistance collision property of the hash function "Edon-F". On the other side, if we have a quasigroup equation containing only one appearance of an unknown x then there is a unique solution of that equation.

Here we propose a hash function which is strongly collision-free. It is called "Edon-F" hash function and it is designed using quasigroup operation on the set A. The idea of using quasigroups as cryptographical tools is elaborated in several papers ([MGB 1999], [MGB 2001], [MK 2000]). As it can be seen from the proposed design "Edon-F" is not only one hash function, but it is rather an infinite family of hash functions. Namely, for any digest size m, an "Edon-F" hash function can be constructed. For the construction we need a quasigroup of order $|A|$ and in the sequel we take that $*$ is a quasigroup operation on the set A. The number of quasigroup operations on a set A with enough large cardinality of A is a huge one. Thus, for $|A|$=256, there are more than 10^{58000} quasigroup operations. So, "Edon-F" can be used as a hash function with key as well, where the key is a quasigroup operation.

2. "Edon-F" HASH FUNCTION

The construction of "Edon-F" hash function will be given in several steps. At first we will define a one-way function $f : A^m \longrightarrow A^m$ that will be enlarged to a hash function $h : A^+ \longrightarrow A^m$. As above, A is a given finite set.

Further on, we will use two auxiliary vectors $U = (u_1, u_2, ..., u_m)$ and $V = (v_1, v_2, ..., v_m)$ that take their values from the set A. During the computational process the vectors will interchange their values. For starting of computations, we initialize the values of V. We take arbitrary elements $b_i \in A$ and we put V to be the vector $V := (b_1, ..., b_m)$.

Let $a_1a_2...a_m$ be a given message of length m, where $a_i \in A$. We compute the values of U as follows. At first, we compute u_1 and u_2 by

$$u_1 := ((a_1 * a_2) * (a_2 * a_1)) * v_1 \qquad (1)$$

$$u_2 := (...((a_1 * a_2) * a_3) * ...) * a_m) * v_2. \qquad (2)$$

Then we compute the values u_i for $i = 3, 4, ..., m$ by the following recurrent process:

$$u_i := (a_i * u_{i-1}) * (a_i * v_i), \qquad (3)$$

where u_{i-1} is the value obtained in the previous step and the values v_i are taken from the vector V.

After that, using the computed values of u_i, we compute the new values of V by the rules of the same kind:

$$\begin{aligned} v_1 &:= ((c_1 * c_2) * (c_2 * c_1)) * u_1 \\ v_2 &:= (...((c_1 * c_2) * c_3) * ...) * c_m) * u_2 \qquad (4) \\ v_i &:= (c_i * v_{i-1}) * (c_i * u_i), \quad i = 3, 4, ..., m \end{aligned}$$

where $c_i = u_i * a_i$, $i = 1, 2, ..., m$.

Now we define the function $f : A^m \longrightarrow A^m$ by $f(a_1...a_m) = v_1...v_m$. The function f will be enlarged to a function $h : A^+ \longrightarrow A^m$ on the following way. At first, any message $\alpha = a_1...a_r \in A^+$ is padded to a message of length mk for some minimal positive integer k. Namely, we fix two different elements b, $c \in A$ and we concatenate the string $bc...c$ to α to obtain a string $a_1...a_rbc...c$ of desired length. So, we may assume that α has length mk for some k, and we present it as a concatenation of substrings of length m:

$$\alpha = a_1...a_m||a_{m+1}...a_{2m}||...||a_{m(k-1)+1}...a_{mk}.$$

We define recursively the function f’ by: $f'(a_1...a_m) = f(a_1...a_m)$ and if $f'(a_{m(i-1)+1}...a_{mi}) = d_1...d_m$ is already computed, we compute

$$f'(a_{mi+1}...a_{m(i+1)}) = f((d_1 * a_{mi+1})(d_2 * a_{mi+2})...(d_m * a_{m(i+1)})).$$

Finally, define $h(a_1...a_r) = f'(a_{m(k-1)+1}...a_{mk})$.

In such a way "Edon-F" hash function is defined.

3. PROPERTIES OF "Edon-F" HASH FUNCTION

Here we show that h has suitable cryptographic properties, i.e. we will prove that h is a one-way strongly collision-free function, and we present experimentally obtained results. At first, we show that f is a one-way function. It is clear that for a given message x, it is easy to compute $f(x)$ using the iterative equations (1), (2), (3) and (4). But, it is computationally infeasible to find an origin $x = x_1 x_2...x_m$ of any given $v \in A^m$. Namely, let $v = v_1v_2...v_m$ be a given string. If we assume that $f(x_1 x_2...x_m) = v_1v_2...v_m$ then in the quasigroup $(A, *)$ we have to solve the following system of equations:

$$\begin{aligned}
&y_1 = ((x_1 * x_2) * (x_2 * x_1)) * b_1 \\
&y_2 = (...((x_1 * x_2) * x_3) * ...) * x_m) * b_2 \\
&y_i = (x_i * y_{i-1}) * (x_i * b_i), && i = 3, 4, ..., m \\
&v_1 = ((z_1 * z_2) * (z_2 * z_1)) * y_1 \\
&v_2 = (...((z_1 * z_2) * z_3) * ...) * z_m) * y_2 && && (5) \\
&v_i = (z_i * v_{i-1}) * (z_i * y_i), && i = 3, 4, ..., m \\
&z_i = y_i * x_i, && i = 1, 2, ..., m
\end{aligned}$$

In this system the values b_i and v_i are known, and x_i, y_i, z_i are unknown for each $i = 1, 2, ..., m$. As it was mentioned above, this system can by solved only by checking. Suppose we have chosen values for unknowns $z_1, z_2,..., z_m$. Then in the system (5) we can compute in unique way the values of $y_1, y_2, ..., y_m$ from the equations

$$\begin{aligned}
&v_1 = ((z_1 * z_2) * (z_2 * z_1)) * y_1 \\
&v_2 = (...((z_1 * z_2) * z_3) * ...) * z_m) * y_2 \\
&v_i = (z_i * v_{i-1}) * (z_i * y_i), && i = 3, 4, ..., m
\end{aligned}$$

and after that the values of $x_1, x_2,..., x_m$ can be computed in unique way from the last equations in (5). But, in such a way obtained values have to satisfy the equations:

$$\begin{aligned}
&y_1 := ((x_1 * x_2) * (x_2 * x_1)) * b_1 \\
&y_2 := (...((x_1 * x_2) * x_3) * ...) * x_m) * b_2. \\
&y_i := (x_i * y_{i-1}) * (x_i * b_i), && \text{for } i = 3, 4, ..., m
\end{aligned}$$

This is a checking system if the values of $z_1, z_2, ..., z_m$ are correctly chosen. For surely finding an origin x (if it exists) such that $v = f(x)$ we have to do $|A|^m$ checkings. Since $|A|^m$ is an exponential function, it is computationally infeasible to find an origin x of any given v, i.e. f is one-way function.

Since the function f' is defined using the function f, it follows that f' is also a one-way function. The proof is the same as the previous one, since in f' we have products $x'_j = d_j * x_j$ instead of x_j, but it does not matter in solving of quasigroup equations. Namely, the mapping $x_j \mapsto x'_j$ is a bijection. So, we can conclude that h is a one-way function.

Let note that the padding function $g(a_1a_2...a_r) = a_1a_2...a_rbc...c$ is an injection from A^+ to A^+. Let $a_1a_2...a_r \neq d_1d_2...d_k$. If $r = k$ then $a_1a_2...a_rbc...c = d_1d_2...d_rbc...c$ implies $a_i = d_i$ for each i=1, 2 , ..., r, a contradiction. If $r < k$ then the element c is on the (k+1)-th position in $a_1a_2...a_rbc...c$ and the element b is on the (k+1)-th position in $d_1d_2...d_kbc...c$, so $a_1a_2...a_rbc...c \neq d_1d_2...d_kbc...c$.

It follows from the previous considerations that the padding cannot be used for obtaining a collision. On the other hand, by using the function h collision cannot be obtained either. Namely, for finding a collision, systems of the kind (5) have to be solved.

We have made many experiments to check the statistical properties of "Edon-F". In Table 1 we present the results of the Hamming distance. We have taken two original strings which differ in only one bit. Then we computed their hash values for different digest sizes $m \in \{40, 64, 80, 128, 160, 256, 320, 384, 512, 1024\}$. We repeated this procedure 1000 times and in the second column of Table 1, the average Hamming distance is given. The third column contains the standard deviation. We can conclude that 1 bit difference in the input message produces $m/2$ differences in the message digest.

Digest size m	Average Hamming distance	Standard deviation
40	20.00	3.16
64	31.99	4.00
80	40.00	4.48
128	63.99	5.65
160	79.88	7.04
256	127.98	8.00
320	159.39	13.27
384	191.42	14.35
512	256.02	11.32
1024	511.98	16.02

Table 1. The Hamming distance

On Table 2 we present the results of the experiments made for a collision. The digest size is chosen to be m = 24, 32, 40, 48 for technical reasons. It can be seen from the Table 2 that for m = 24, a collision was found when hash values of 3258 randomly chosen messages were taken. For m = 48, 5 960 000 messages were needed for obtaining a collision. On the third column of the Table 2 the values of $2^{m/2}$ are given. We note that a collision is obtained for less than $2^{m/2}$ messages, i.e. in approximately $2^{m/2-2}$ messages. Considering the Birthday attack on hash functions we can see that "Edon-F" hash function has smaller security. (This means that one have to take somewhat larger values of m, i.e. to take m+4 instead of m.)

Digest size m	Number of messages for obtaining a collision	$2^{m/2}$
24	3258	4096
32	15170	65536
40	375268	1048576
48	5960000	16777216

Table 2. Number of messages for obtaining a collision

The complexity of our algorithm is linear. Namely, for one round of computing the function f', one needs $10m - 4$ applications of the operation $*$. So, for input message of length r, one needs to apply $10r$ times the operation $*$. We conclude that the complexity of "Edon-F" hash function is $O(r)$ where r is the length of the input message.

4. CONCLUSION

The everyday enlargement of the computer performances arise the problems of security of the cryptographical products. Considering the hash functions, digest sizes have to be larger and larger. Here we propose the hash function "Edon-F" having a digest size of variable length. So, "Edon-F" is an adoptable hash function that can follow the changes of computer technologies. It is built by using properties of quasigroups and its collision resistant security is mathematically proved. It is of linear complexity of the length of the input message, so it is fast one. By the experiments we could see that it is a good mixture and it has some weakness considering a Birthday kind of attacks which can be avoid by choosing somewhat larger digest size.

REFERENCES

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

[GMB 2003] Gligoroski, D., Markovski, S. and Bakeva, V.: "Edon–C" and "Edon–R" – two infinite classes of strongly collision resistant hash functions with variable length of output (*preprint*)

[MGB 1999] Markovski, S., Gligoroski, D., and Bakeva, V.: Quasigroup String Processing: Part 1, *Contributions, Sec. Math. Tech. Sci., MANU, XX 1-2(1999) 13-28.*

[MGB 2001] Markovski, S., Gligoroski, D., and Bakeva, V.: Quasigroup and Hash Functions, *Disc. Math. and Appl., Sl.Shtrakov and K. Denecke ed., Proceedings of the 6th ICDMA, Bansko 2001, 43-50*

[MK 2000] Markovski, S., Kusakatov, V.: Quasigroup String Processing: Part 2, *Contributions, Sec. math. Tech.Sci., MANU, XXI, 1-2(2000) 15-32*

[MOV 1997] A. Menezes, P. van Oorschot, and S. Vanstone: *Handbook of Applied Cryptography*, CRC Press, Inc., 1997

[PBD 1997] B. Preneel, A. Bosselaers, H. Dobbertin: The cryptographic hash function RIPEMD-160, *CryptoBytes, Vol. 3, No. 2, 1997, 9-14.*