Behavioral and Security Study of the OHFGC Hash Function

Ahmed Drissi and Ahmed Asimi
(Corresponding author: Ahmed Drissi)

Department of Mathematics, Faculty of Sciences, Ibn Zohr University, Agadir, Morocco
B.P 8106, Agadir, Morocco
(Email: idrissi2006@yahoo.fr)

(Received Mar. 10, 2016; revised and accepted May 22 & June 10, 2016)

Abstract

The designs of several hash functions (SB, FSB, RFSB, SFSB, OHFGC, $\cdots$) are based on the error-correcting codes properties. The hash function based on the classical Goppa code "OHFGC" [7] is distinguished by the possibility that an user selects certain parameters to achieve a level of performance and security corresponds to its needs. The objective of this article examines the security features of the hash function "OHFGC" and its behavior in order to propose relevant parameters for different user situations. We also propose both a method to summarize all parameters in one and a method that links the size of the hashed to the document.

Keywords: Classical Goppa code, one way hash function, syndrome decoding

1 Introduction

Several hash functions (SB, FSB, RFSB, SFSB, OHFGC, $\cdots$) [2, 3, 4, 9] are based on the error-correcting codes properties. The hash function based on the classical Goppa code "OHFGC" [7] is distinguished by the ability that an user selects certain parameters to achieve a level of performance and security corresponds to its needs. In the next section, we recall the algorithms components of the OHFGC. Section Three is devoted to the security study by the design model and the hashed size. In section four, we study the performance, the behavior of the OHFGC and its sensitivity to initial conditions. Our proposed method of choosing a single parameter from the others is presented in section five. It ends with a conclusion. Table 1 is the notations used in this paper.

Let the finite field $F_{2^m} = \{0, 1, \alpha, \alpha^2, \cdots, \alpha^{2^m-2}\}$ and its primitive element α which is the root of a primitive polynomial of the degree m on F_2 [10]. There is a biunivocal correspondence between the elements of F_{2^m}, as a F_2 vector space its base is $(1, \alpha, \alpha^2, \cdots, \alpha^{m-1})$, and the elements of F_{2^m} is defined by:

$$\begin{array}{rcl} \varphi : F_{2^m} & \longrightarrow & F_{2^m} \\ x = \sum_{i=0}^{i=m-1} a_i \alpha^i & \longrightarrow & (a_0, \cdots, a_{m-1})^T \end{array}$$

2 Recall on the OHFGC Hash Function

The hash of a message M by $OHFGC$ is according to the MERKLE and DAMGARAD model [6, 11], in the heart of this model there is a compression function. The compression function of the $OHFGC$ [7] is composed of the following algorithms:

- A compression function CF.

  A compression function CF, of the input size n and of the output r, based on H (a parity check matrix of a classical Goppa code), and is defined as follows:

$$\begin{array}{rcl} CF : F_{2^n} & \rightarrow & F_{2^r} \\ x & \rightarrow & x^{(1)} + H\phi(x)^t, \end{array}$$

  with $x = (x^{(2)}, x^{(1)})$, $x^{(1)} \in F_{2^{n-r}}$, $x^{(2)} \in F_{2^r}$ and

$$\begin{cases} \phi(x) = x & \text{if } w(x) \leq \frac{n}{2} \\ \phi(x) = x \oplus 1^n & \text{if } w(x) \succ \frac{n}{2} \end{cases}$$

- The generation of a parity check matrix.

  The generation of a parity check matrix ($H \in M_{r,n}(F_2)$) from a primitive element of a field F_{2^m} and an integer n with $(2m \prec n \prec 2^m - 1)$.

The generation of H is done as follows:

1) Choose an integer n such as $2m \prec n \prec 2^m - 1$ and a primitive element α of F_{2^m}.

2) Calculate $(i_j)_{j=1}^n$ with $i_j = n^j \bmod (2^m - 1)$ and $t = E(\frac{n}{2m})$.

Table 1: Notations

N	The set of integers.
$F_2 = \{0, 1\}$	A finite field of the two elements.
n	an integer.
m	an integer.
F_{2^m}	The finite field of 2^m elements, with m an integer.
${F_{2^m}}^*$	The multiplicative group of the nonzero elements of F_{2^m}.
$M_{rxn}(K)$	The set of rxn matrices with coefficients in an abelian field K.
${F_2}^n$	The set of the vectors that components 0 or 1 and their length is n.
$1^n = (1, 1, ..., 1)$	The vector of n components equal to 1.
$OHFGC$	One-way Hash function synchronized based on Goppa Codes.
CF	Compression function.
$E(x)$	The integer part of x.
t	An integer.
$\Gamma(L, x^t)$	A classical Goppa code with L its support and x^t its polynomial.
$OHFGC(m)$	One-way Hash function based on Goppa Code with his principle parameter m.
$w(x)$	The sum of the components of x.
$\oplus$	An XOR operation.

3) Calculate $K' = (\varphi(\alpha_j^{i-t-1}))_{i=1,\cdots,t;j=1,\cdots,n}$.

4) The parity control matrix H is composed of lines in K' without repetition and in the same order. This is the parity check matrix of $\Gamma(L, x^t)$ in F_2 of rxn type.

5) r is the output size of $OHFGC$ and the compression function CF.

Remark 1. *We cannot predict the value of the hashed size r before the construction of H, this is due to a particular property of the parity check matrix of a classical Goppa code. We have to recourse to implementation.*

3 The OHFGC Security Study

The security of the entire hash functions depends mainly on its design model and the hashed size. The first ensures resistance against structural attacks and the second guarantees its resistance to generic attacks. In the two following paragraphs we discuss these principles in the case of the OHFGC.

1) The OHFGC security based on design model.

   The OHFGC is built according to the model MERKLE and DAMAGARAD [6, 11]. MERKLE [6] showed that the security of any hash function is designed according to the model is summarized in compression function of the resistance, constructed with, at the three security criteria (resistance to pre-image, second pre-image and collisions).

   For hash functions were based on code, including the OHFGC, the security is easily linked to the difficulty of the problem by decoding syndromes [4, 8, 12].

   The following two issues proved hard [7], provide the security for the OHFGC.

   Given H a matrix of the type rxn of elements of the F_2 and $s \in F_{2^r}$.

   Find $x = (x^{(2)}, x^{(1)}) \in F_{2^{n-r}} x F_{2^r}$ such as $x^{(1)} + Hx^t = s$.

   Given H a matrix of the type of rxn of elements of the F_2 and $s \in F_{2^r}$.

   Find $x = (x^{(2)}, x^{(1)}) \in F_{2^{n-r}} x F_{2^r}$ and $y = (y^{(2)}, y^{(1)}) \in F_{2^{n-r}} x F_{2^r}$ such as $x^{(1)} + y^{(1)} = H(x + y)^t$.

2) The OHFGC security based on its hashed size.

   Generic attacks [5] (see Table 2) depend on the number of the possible hashed 2^r of the size r. As to ensure safe of some functions hash, simply increase the size of hashed (at the moment the sizes 256 and 512 are considered acceptable). For the OHFGC, we propose to give varying sizes included in intervals depending on its primary endpoint: primitive polynomial. In addition, it is distinguished by the possibility of extending these intervals by increasing the degree of the primitive polynomial. This property gives the complexity of the OHFGC for a longer time.

Table 2: Complexity of the best generic attacks

Generic attack	Complexity
Search pre-image	2^r
Research of second pre-image	2^r
Research of collisions	$2^{\frac{r}{2}}$

4 The Behavior Study and the Performance of OHFGC Function

The parameters of the OHFGC function are n, m, α (α is a root of a primitive polynomial $p(x)$ of degree m) [1] and hashed size r. Tables 3, 4, 5, 6, 7, provides examples of the parameters that can be used. These examples give us an idea of the possible choices.

Table 3: The hashed size for m=8

m	$p(x)$	n	Hashed size r	2^m
8	$x^8+x^5+x^3+x+1$	254	4	256
8	$x^8+x^5+x^3+x+1$	253	90	256
8	$x^8+x^5+x^3+x+1$	252	120	256
8	$x^8+x^5+x^3+x+1$	251	16	256
8	$x^8+x^5+x^3+x+1$	250	120	256
8	$x^8+x^5+x^3+x+1$	249	120	256
8	$x^8+x^5+x^3+x+1$	248	120	256
8	$x^8+x^5+x^3+x+1$	247	90	256
8	$x^8+x^5+x^3+x+1$	100	43	256

Table 4: The hashed size for m=9

m	$p(x)$	n	Hashed size r	2^m
9	x^9+x^5+1	254	4	512
9	x^9+x^5+1	253	90	512
9	x^9+x^5+1	510	4	512
9	x^9+x^5+1	509	251	512
9	x^9+x^5+1	508	246	512
9	x^9+x^5+1	507	251	512
9	x^9+x^5+1	506	252	512
9	x^9+x^5+1	504	252	512
9	x^9+x^5+1	503	59	512
9	x^9+x^5+1	502	60	512
9	x^9+x^5+1	501	243	512
9	x^9+x^5+1	500	243	512
9	x^9+x^5+1	400	198	512
9	x^9+x^5+1	300	52	512
9	x^9+x^5+1	200	99	512

These data lead us to seek to have a OHFGC function of the variable hashed size and summarize the parameters in one.

1) The behavior study of the OHFGC.

   Any modification of the hashed document leads a variation on the hashed. The variation on the hashed is measured by the Hamming distance between the two vectors (hashed). The graphs (Figures 1, 2, 3, 4) represent the Hamming distance between the

Table 5: The hashed size for m=10

m	$p(x)$	n	Hashed size r	2^m
10	$x^{10}+x^3+1$	1022	495	1024
10	$x^{10}+x^3+1$	1021	373	1024
10	$x^{10}+x^3+1$	1020	510	1024
10	$x^{10}+x^3+1$	1019	364	1024
10	$x^{10}+x^3+1$	1018	500	1024
10	$x^{10}+x^3+1$	1017	500	1024
10	$x^{10}+x^3+1$	1016	500	1024
10	$x^{10}+x^3+1$	1015	365	1024
10	$x^{10}+x^3+1$	1014	500	1024
10	$x^{10}+x^3+1$	1013	500	1024
10	$x^{10}+x^3+1$	1012	493	1024
10	$x^{10}+x^3+1$	1011	500	1024
10	$x^{10}+x^3+1$	1000	387	1024
10	$x^{10}+x^3+1$	100	50	1024
10	$x^{10}+x^3+1$	800	400	1024

Table 6: The hashed size for m=11

m	$p(x)$	n	Hashed size r	2^m
11	$x^{11}+x^2+1$	2046	4	2048
11	$x^{11}+x^2+1$	2045	1011	2048
11	$x^{11}+x^2+1$	2044	1012	2048
11	$x^{11}+x^2+1$	2043	1011	2048
11	$x^{11}+x^2+1$	2042	1012	2048
11	$x^{11}+x^2+1$	2041	1012	2048
11	$x^{11}+x^2+1$	2040	1012	2048
11	$x^{11}+x^2+1$	2039	1011	2048
11	$x^{11}+x^2+1$	2038	1012	2048
11	$x^{11}+x^2+1$	2037	1012	2048
11	$x^{11}+x^2+1$	2000	990	2048

Table 7: The hashed size for m=12

m	$p(x)$	n	Hashed size r	2^m
12	$x^{12}+x^6+x^4+x+1$	4094	4	4096
12	$x^{12}+x^6+x^4+x+1$	4093	1500	4096
12	$x^{12}+x^6+x^4+x+1$	4092	1783	4096
12	$x^{12}+x^6+x^4+x+1$	4091	64	4096
12	$x^{12}+x^6+x^4+x+1$	4090	1602	4096
12	$x^{12}+x^6+x^4+x+1$	4089	1767	4096
12	$x^{12}+x^6+x^4+x+1$	4088	1587	4096
12	$x^{12}+x^6+x^4+x+1$	4087	16	4096
12	$x^{12}+x^6+x^4+x+1$	4000	64	4096
12	$x^{12}+x^6+x^4+x+1$	3000	128	4096
12	$x^{12}+x^6+x^4+x+1$	409	194	4096

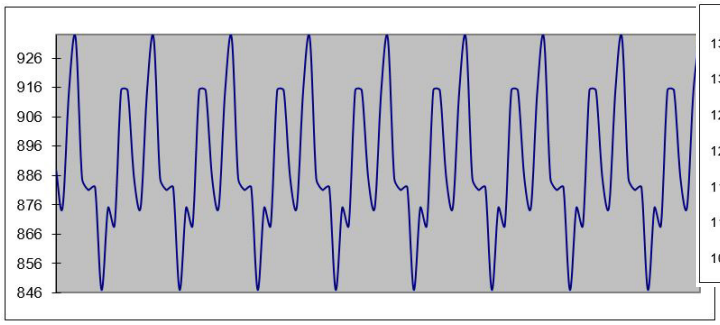

Figure 1: OHFGC (4092, 12,1783)

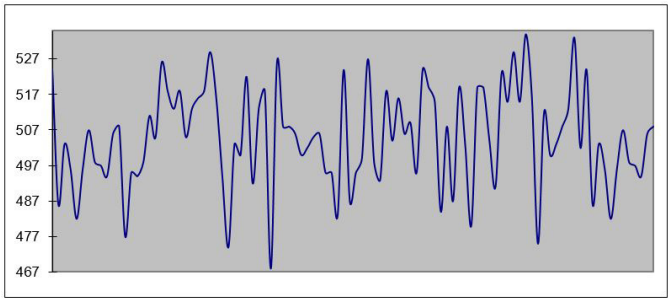

Figure 2: OHFGC(2040,11,1012)

hashed of the original document and the hashed of the amended document by a single bit within the first 100 positions in the original document.

In summary, in the four examples of the OHFGC function, each modification of the document to hash, by a single bit, causes variation of the hashed by approximately half the number of bits.

2) The OHFGC performance.

We hashed a file of size 1.01 MB by the OHFGC(n, m, r) function, Table 8 shows the execution time for the chosen parameters and which have its performance.

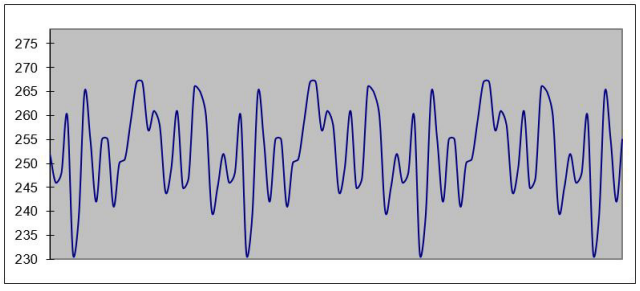

Figure 3: OHFGC (1020, 10,510)

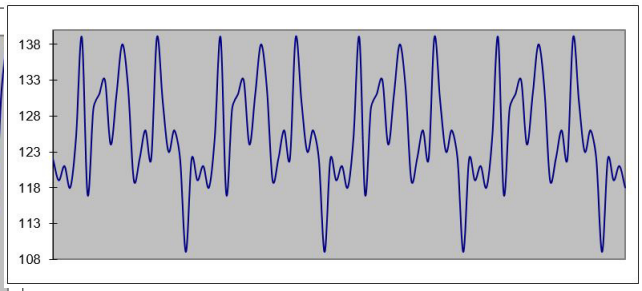

Figure 4: OHFGC (504, 9,252)

Table 8: Performance of the on core (TM) 2 duo CPU 2.00 GHZ

functions	Execution time
OHFGC (4092, 12,1783)	10,98200 s
OHFGC (2040, 11,1012)	6,70800 s
OHFGC (1020, 10,510)	3,52600 s
OHFGC (504, 9,252)	1,95000 s

5 Proposal Method for Selecting Parameters

After the behavioral study of the $OHFGC(n, m, r)$, we propose to keep a single parameter of the $OHFGC(m)$ and to link n to the document size to be hashed by the relation $n = (2m+1+document\ size) mod(2^m-2)$ and by following the hashed size r will vary from one document to another in the interval $[1,mE(\frac{2^m-2}{2m})]$.

Explication 1. *The hashed size is between 1 and* $[1,mE(\frac{2^m-2}{2m})]$ *indeed. The matrix H has at least one line. we have* $n = (2m+1+document\ size) mod(2^m-2)$ *then* $2m \prec n \prec 2^m-2$. *We have also* $r \leq mt$ *(since r is the number of lines in H after reduction) consequently* $1 \leq r \leq mE(\frac{2^m-2}{2m})$.

Remark 2. *Having the variable hashed size in a range increase the complexity of generic attacks. We take for example the following intervals (Table 9).*

Table 9: Examples of the intervals document size

m	$[1,mE(\frac{2^m-2}{2m})]$
8	[1,120]
9	[1,252]
10	[1,510]
11	[1,1023]
12	[1,2040]

6 Conclusion

In conclusion, we can announce that our OHFGC(m) function parameterized by a primitive polynomial of the degree m and of the variable size from one document to another, is an efficient and secure function. The flexibility of choosing the parameter m of the OHFGC depending on the context of the use ensures that our exclusive function can last longer as it will be used by different users in different contexts.

References

[1] A. Asimi and A. Lbekkouri, "Determination of irreducible and primitive polynomials over a binary finite field," 2009. (`file:///C:/Users/user/Downloads/asimiprim.pdf`)

[2] D. Augot, M. Finiasz, P. Gaborit, S. Manuel, and N. Sendrier, "SHA-3 proposal: FSB," Submission to NIST, pp. 81–85, 2008.

[3] D. Augot, M. Finiasz, and N. Sendrier, "A family of fast syndrome based cryptographic hash functions," in *Progress in Cryptology (Mycrypt'05)*, pp. 64–83, Springer, 2005.

[4] D. J. Bernstein, T. Lange, C. Peters, and P. Schwabe, "Really fast syndrome-based hashing," in *Progress in Cryptology (AFRICACRYPT'11)*, pp. 134–152, Springer, 2011.

[5] C. Boura, *Analyse De Fonctions De Hachage Cryptographiques*, Ph.D. Thesis, University Pierre et Marie Curie-Paris VI, 2012.

[6] I. B. Damgard, "A design principle for hash functions," in *Advances in Cryptology (CRYPTO'89)*, pp. 416–427, Springer, 1989.

[7] A. Drissi and A. Asimi, "One-way hash function based on goppa codes ohfgc," *Applied Mathematical Sciences*, vol. 7, no. 143, pp. 7097–7104, 2013.

[8] M. Finiasz, "Nouvelles constructions utilisant des codes correcteurs derreurs en cryptographie á clef publique," These de doctorat, *É*cole Polytechnique, 2004.

[9] W. R. Ghanem, M. Shokir, and M. Dessoky, "Defense Against Selfish PUEA in Cognitive Radio Networks Based on Hash Message Authentication Code," *International Journal of Electronics and Information Engineering*, vol. 4, no. 1, pp. 12–21, 2016.

[10] R. Lidl and H. Niederreiter, *Finite Fields (Encyclopedia of Mathematics and Its Applications, vol. 20)*, Reading, MA, USA: AddisonWesley, pp. 428–431, 1983.

[11] R. C. Merkle, "One way hash functions and des," in *Advances in Cryptology (CRYPTO'89)*, pp. 428–446, Springer, 1989.

[12] N. Sendrier, *Cryptosyst Emes a Cl e Publique Bas es Sur Les Codes Correcteurs D'erreurs*, Habilitation diriger les recherches, Universit Pierre et Marie Curie, Paris, France (in French), 2002.

Ahmed Drissi received his PhD degree in cryptology from the Faculty of Science, the University Ibn Zohr Agadir, Morocco in 2014. His research interests include Code theory and the Cryptology.

Ahmed Asimi received his PhD degree in Number theory from the University Mohammed V Agdal in 2001. He is reviewer at the International Journal of Network Security (IJNS). His research interest includes Number theory, Code theory, and Computer Cryptology and Security. He is a full professor at the Faculty of Science at Agadir since 2008.