Applied Mathematical Sciences, Vol. 7, 2013, no. 143, 7097 - 7104
HIKARI Ltd, www.m-hikari.com
http://dx.doi.org/10.12988/ams.2013.311667

One-way Hash Function Based on Goppa Codes « OHFGC »

Ahmed DRISSI and Ahmed ASIMI

LabSiv : Laboratoire des Systèmes informatiques et Vision
ESCCAM : Equipe de la Sécurité, Cryptographie,
Contrôle d'Accès et Modélisation
Department of Mathematics, Faculty of Sciences
Ibn Zohr University, Agadir, Morocco
idrissi2006@yahoo.fr, asimiahmed2008@gmail.com

Abstract

The design of family of hash function based syndrome and its variants are based on a compression function, that uses a parity check matrix of an error correcting code (random, quasi-cyclic).
The objective of this paper is to propose a one-way hash function (OHFGC) of variable size based on classical Goppa codes and the scheme of MERKLE and DAMGARAD. Classical Goppa codes, of pseudo-random characteristic, for regenerating parity check matrix from a primitive element of a finite field F_{2^m} and an integer $n > 2m$, and for the design of the compression function. The scheme of MERKLE and DAMGARAD [4,5,6] whose the compression function is based on the calculation of syndromes, the initial vector specific to each message and the reduction function of weight of a given word.

Keywords: One way Hash function, Syndrome decoding, Classical Goppa codes, Compression function and Parity check matrix.

I.) Introduction and Notations

In the following we note:
N : The set of natural integers.

$F_2 = \{0,1\}$: The finite field of two elements

F_{2^m} : The finite field of 2^m elements, with m an integer.

$F_{2^m}^*$: The group of non-zero elements F_{2^m}.

$M_{r\times n}(K)$: The set of matrix of type $r \times n$ with coefficients in an abelian field K.

F_2^n : The set of vector of components 0 or 1 and of length n.

$1^n = (1,1,...,1)$: Vector of n components equal to 1.

$OHFGC$: One-way Hash function based on Goppa Codes.

CF : Compression function.

|| : Concatenation.

$E_r(M)$: Function of extraction of r first message bits M.

$E(x)$: The integer part of x.

$\Gamma(L, g(x))$: The classical Goppa code of support L and of the generator polynomial $g(x)$.

:= : Affectation.

A hash function is a function that takes as input an word of variable size and returning output an word of fixed and predetermined size. These functions, which are one of the most studied subjects currently cryptography, are very useful especially in the field of databases in the case of Web applications to prove user authentication web server in and the exchange of confidential data signed by a hash function guaranteeing their integrity.

Historically, Shor [7] showed in 1994 that quantum computers can break cryptosystems based on the factoring problem and discrete logarithm. Since it can factor integers and extract discrete logarithms. This is why post-quantum cryptosystems public key including cryptographic hash functions to be safe during the onset of quantum computers must be invented. Cryptosystems based on error-correcting codes are among the best known and most studied schemes. Their security relies on the problem of decoding a random code. In 2003 the first design of hash function based on the theory of codes was proposed by Augot et al [2]. His proposal is based on the construction of MERKLE DAMGARAD. A compression function extracted by the syndrome decoding problem whose it requires a matrix of random parity and an algorithm that incorporates a word given weight. In 2005 the same authors proposed an improvement by introducing regular coding [2]. And to reduce memory consumption and increase speed, Finiasz et al [3] introduced in 2007 quasi-cyclic codes instead of random codes. His proposal called FSB Fast hash function on the syndrome is still too long and needs improvement. An improved variant of FSB in terms of efficiency and storage capacity is given by Bernstein [4] in 2011 under the name RFSB, which is based on a parameterized code.

The security of a hash function constructed with the scheme MERKLE and DAMGARD proved at least as high as the security of the compression function

used, so it is necessary to prove for the compression function. Since resistance to second pre-image is strictly weaker than resistance to the collisionWe check that the compression function is collision resistant and inversion. Which is easily linked to the difficulty of the two syndromes decoding problems following (see section IV) :

« Given H a matrix of type $r \times n$ and of elements of F_2 and $s \in F_2^r$. Find $x = \left(x^{(2)}, x^{(1)}\right) \in F_2^{n-r} \times F_2^r$ as $x^{(1)} + Hx^t = s$ ».

« Given H a matrix of type $r \times n$ and of elements of F_2 and $s \in F_2^r$. Find $x = \left(x^{(2)}, x^{(1)}\right) \in F_2^{n-r} \times F_2^r$ and $y = \left(y^{(2)}, y^{(1)}\right) \in F_2^{n-r} \times F_2^r$ as $x^{(1)} + y^{(1)} = H(x+y)^t$. »

These difficulties have pushed us and encouraged, in this article, to design a hash function one-way variable size based on twice schemes. Firstly, classical Goppa codes, pseudorandom characteristic which requires pre-implementation binary finite fields, for regenerating parity check matrix from a primitive element of a field F_{2^m} and an integer $n > 2m$ and for the design of compression functions and weight reduction of a given word whose aim is to reduce the number of XORs in the calculations of syndromes. Secondly, the scheme of MERKLE and DAMGARAD [4,5,6].

Our work is organized as follows: we begin by recalling the construction of MERKLE DAMGARD in section two and binary finite fields in Section Three. The interesting property pseudo-random of classical Goppa code is modelled by difficult problems in section four. In section five we give a detailed description of our proposed hash function. And we finish with a conclusion.

II.) Model of MERKLE DAMGARAD

The construction of most functions of cryptographic hash follows the same model presented by MERKLE and DAMGARAD [4,5,6] in 1989. The core of this scheme is the compression function. The principle of this model is:

Given a message M and a compression function his input size is n and output size r and an initial vector $h_0 = IV$ its size r.

1.) Segmenting M into s blocks M_i of the same size $n-r$. If the document length is not a multiple of $n-r$, we add 0 (padding) at the end of the document so that its size is a multiple of $n-r$.
2.) $h_i = CF\left(M_i \,\|\, h_{i-1}\right)$ for $i \in \{1,...,s\}$.
3.) h_s is the hash of the message M.

III.) Binary finite field

The design of our hash function requires a pre-implementation of a binary finite field. The finite field is generally constructed from primitive polynomials. In [1],

the second author of this article has given an algorithm for recovering all primitive polynomials over a binary finite field of a given degree.
The following results characterize the irreducible polynomials over a binary field of a fixed degree. The proof is well-known, see for example [11], [12] and [13].

Definition 1. A polynomial $f(x) \in F_2[x]$ is said to be irreducible over F_2 if $f(x)$ has a positive degree and every factorization of $f(x)$ in $F_2[x]$ must involve a constant polynomial.

Corollary 1. Let $f(x) \in F_2[x]$ be an irreducible polynomial over F_2 of degree m. Then the splitting field of f over F_2 is given by F_{2^m}.

Definition 2. Let $f(x) \in F_2[x]$ be a nonzero polynomial. If $f(0) \neq 0$, then the least positive integer e for which $f(x)$ divides $x^e - 1$ is called the order of f and denoted by $ord(f) = ord(f(x))$.

Theorem 1. Let $f(x) \in F_2[x]$ be an irreducible polynomial over F_2 of degree m and with $f(0) \neq 0$. Then $ord(f)$ is equal to order of any root of f in the multiplicative group F_{2^m}.

Definition 3. A polynomial $f(x) \in F_2[x]$ of degree m is a primitive polynomial over F_2 if it is a minimal polynomial over F_2 of a primitive element of F_{2^m}.

Theorem 2. A polynomial $f(x) \in F_2[x]$ of degree m is a primitive polynomial over F_2 if and only if f is monic, $f(0) \neq 0$ and $ord(f) = 2^m - 1$.

IV.) The pseudo-random Character of Goppa code

Definition 4 : The Classical Goppa code (rational) of support $L = (\alpha_1, \alpha_2, ..., \alpha_n) \in F_{2^m}^n$ and Goppa polynomial $g(x) \in F_{2^m}[x]$ of a degree r noted $\Gamma(L, g)$ is defined by $\Gamma(L, g) = \{a = (a_1, a_2, ..., a_n) \in F_2^n / Ha^t = 0\}$ with $H = \left(\alpha_j^{i-1} g(\alpha_j)^{-1}\right)_{\substack{i=1,...,r \\ j=1,...,n}}$.

H is said parity check matrix $\Gamma(L, g)$.

The classical Goppa code has a pseudo-random interesting property, formalized by two difficult problems: One, which has been proved NP-hard by M.Finiasz in [9], is a special case of decoding by syndrome and other, that was shown average difficult by N.Sendrier in [10], this indistinguishability of a Goppa code from a random code.

IV.1.) The syndrome decoding problem

So far there is no efficient algorithm to decode the code in random. This open

problem is formalized as follows:
« Given a matrix H of $M_{r\times n}(F_2)$, s a vector of F_2^r and $\omega \in \mathrm{N}$. Find $x \in F_2^n$ as $Hx^t = s$ and $\omega(x) \le \omega$ ».

IV.2.)The problem of decoding bounded of Goppa code

Given a binary matrix H of type $n-k\times n$ (the parity check matrix of a Goppa code $[n,k]$)

and a syndrome $s \in F_2^{n-k}$. Find an word $e \in F_2^n$ as $\omega(e) \le \dfrac{n-k}{\log_2(n)}$ and $He^t = s$.

IV.3.)The problem of indistinguishability of a Goppa code

The property pseudo-random of Goppa code is related to the indistinguishability of the model code as follows:
Given a binary matrix H of type $n-k\times n$. Check if H is a parity check matrix of a Goppa code $[n,k]$ or not.

V.) Our proposal Function OHFGC

In this article we give the design of a cryptographic hash function, denoted OHFGC, his size of output is variable and depends only on the structure of the regenerator. The size of its output is variable and depends only on the structure of the regenerator and control parity check matrix.

The design principle of our proposal is as follows:

1.) Regeneration of a parity check matrix $\left(H \in M_{r,n}(\mathrm{F}_2)\right)$ from a primitive element of a field F_{2^m} and an integer $n > 2m$.
2.) The design of a reduction function of weight of a word.
3.) Design compression function (CF), of input's size n and the output's size r, based on H.
4.) To hash a message M by $OHFGC$, we proceed as follows:
   4.1) Segmenting M into s blocks M_i of the same size $n-r$. If the document length hashed is not a multiple of $n-r$, we add 0 (padding) at the end of the document so that its size is a multiple of $n-r$.
   4.2) Calculate initial vector $h_0 = IV = E_r(M)$.
   4.3) Calculate $h_i = CF(M_i \| h_{i-1})$ for $i \in \{1,...,s\}$.
5.) h_s is the hash of the message M $\left(h_s = OHGC(M)\right)$.

V.1) Generation of a parity check matrix

Let $F_{2^m} = \left\{0,1,\alpha,\alpha^2,...,\alpha^{2^m-2}\right\}$ be the finite field of 2^m elements with α a primitive generator of the group $F_{2^m}^*$. It was a correspondence between the

elements of F_{2^m}, as F_2 vector space its basis $\left(1,\alpha,...,\alpha^{m-1}\right)$, and elements of F_2^m defined by:

$$\begin{array}{ccccc}\varphi & : & \mathrm{F}_{2^m} & \to & \mathrm{F}_2^m \\ & & x=\sum_{i=0}^{m-1} a_i\alpha^i & \mapsto & \left(a_0,...,a_{m-1}\right)^T\end{array}$$

To generate a parity check matrix of our proposal, we proceed as follows:

1.) We choose an integer $n > 2m$ and $n \neq 2^m - 1$ and a primitive element α of F_{2^m}.

2.) We calculate $\left(i_j\right)_{j=1}^{n}$ where $i_j = n^j \bmod(2^m - 1)$.

3.) We put $t = E\left(\dfrac{n}{2m}\right)$.

4.) We set $\alpha_1 = \alpha^{i_1}, \alpha_2 = \alpha^{i_2}, ..., \alpha_n = \alpha^{i_n}$ and $L = \left(\alpha_1, \alpha_2, ..., \alpha_n\right)$.

5.) We put $K = \left(\alpha_j^{i-t-1}\right)_{\substack{i=1,...,t \\ j=1,...,n}}$ the parity check matrix of $\Gamma\left(L, x^t\right)$ in F_{2^m}.

we calculate $K' = \left(\varphi\left(\alpha_j^{i-t-1}\right)\right)_{\substack{i=1,...,t \\ j=1,...,n}}$. The parity check matrix H is composed of lines of K' without repetition and in the same order. This is the parity check matrix of $\Gamma\left(L, x^t\right)$ in F_2 of type $r \times n$.

r is the output size of *OHFGC* and of compression function *CF*.

Proposition : The size of compression r of *OHFGC* does not exceed $\dfrac{n}{2}$ $\left(r \leq \dfrac{n}{2}\right)$.

Proof

The number of lines of K' is $m \times t$ and of H is r, then $r \leq m \times t$.

Or we have $t = E\left(\dfrac{n}{2m}\right)$, therefore $t \leq \dfrac{n}{2m}$ (i.e $mt \leq \dfrac{n}{2}$), whence $r \leq \dfrac{n}{2}$.

V.2) The function of weight reduction of a vector

The objective of this section is to reduce the weight of a given word.

Our compression function $CF: F_2^n \to F_2^r$, with $n > r$, that is just a multiplication of a matrix type $r \times n$ by a vector of length n and of weight w. The computational complexity is only the number of XORs of columns of this matrix. To decrease, it is sufficient to reduce weight w.

Proposition 1 : Let $x \in F_2^n$ and $y = (x + 1^n) \bmod(2)$, then it was $w(y) = n - w(x)$. In this case it has to be $w(x) \leq \dfrac{n}{2}$ or else $w(y) \leq \dfrac{n}{2}$.

Proof

To obtain $(x + 1^n) \bmod(2)$ from x, we swap the 1's with 0's in the word x.

Therefore if the number of 1 is w then the number of 0 in x is $n-w$.

Show that $w \le \frac{n}{2}$ or else $n-w \le \frac{n}{2}$. Otherwise $w > \frac{n}{2}$ and $n-w > \frac{n}{2}$, whence $n > n$ which is absurd.

This proposal gives us the following technique to reduce weight:

Definition: The weight-reducing function is defined by:

$\phi : F_2^n \to F_2^n$

$$x \to \phi(x) = \begin{cases} x & si \quad w(x) \le \frac{n}{2} \\ x \oplus 1^n & si \quad w(x) > \frac{n}{2} \end{cases} .$$

V.3) Compression function

Our compression function CF is defined by:

$$CF : F_2^n \to F_2^r$$
$$x \to x^{(1)} + H\phi(x)^t$$

with $x = (x^{(2)}; x^{(1)})$, $x^{(1)} \in F_2^r$ and $x^{(2)} \in F_2^{n-r}$.

The compression of $x \in F_2^n$ will be as follows:

1) We calculate the weight of x, ($w := \sum_{i=1}^{n} x_i$, $x = (x_1, x_2, ..., x_n)$).

2) If $w \le \frac{n}{2}$ then $y := x$, otherwise $y_i = (x_i + 1) \bmod(2)$ for $i = 1, ..., n$

3) $z := E_r(y)$

4) $s := (z + Hy^t) \bmod(2)$

VI.) Conclusion

We have presented a new variant of a hash function based on the syndrome decoding. This variant uses the generation of a parity check matrix of a Goppa code rational, which is among codes the most widely used in cryptography and has a pseudo-random character. Our proposal is promising, indeed it is more than resist quantum computers, and it has a variable size that meets our needs to choose the size of the hash we desire and increases safety.

References

[1] ASIMI Ahmed 2013. Determination of irreducible and primitive polynomials over a binary finite field (submitted).

[2] D.Augot, M.Finiasz and N.Sendrier. A family of fast syndrome based cryptographic hash functions. In E. Dawson and S. Vandeny eds. My crypt 2005 springer LNCS (2005)
[3] D.Augot, M.Finiasz, Gaborit,S.Manuel and N.Sendrier SHA-3 proposal :FSB submission to the SHA-3 NIST competition 2008
[4] Daniel J. Berstein, Tanja Lange, Christiane Peters and Peter Schawabe. Really fast syndrome-based hashing. In AFRICA CRYPT 2001,lecture notes in computer science,vol6737 springer-verlag berlin heidelberg 2011.
[5] R.C. Merkle. One way Hash functions and DES. in Gilles Brassard, editor advances in cryptology –CRYPTO'89 proc volume 435 Springer 1990
[6] I. Dangard. Design Principle for hash functions. In Gilles Brassard, editor advances in cryptology –CRYPTO'89 proc volume 435 Springer 1990
[7] P.W.Shor. Aldorithms for quantum computation : discrete logarithme and factoring. In SFC'94 proc of the 35th annuel symposium on foundations of computer science. IEEE computer society (1994)
[8] Report cryptography lab SS 2011 Implementation of the RFSB hash function Lucas Rothamel ,Manuel weil Darmstadt,Germany
[9] Mathieu Finiasz.2004. Nouvelles constructions utilisant des codes correcteurs d'erreurs en cryptographie à clé publique. Thèse INRIA.
[10] Sendrier :crypyosystemes à clé publique basé sur les codes correcteurs d'erreurs.Habilation à diriger des recherches,INRIA,2002.
[11] R. Lidl, H. Niederreiter, Finite fields, Encyclopedia of math, and its Appl, vol 20, Addison- Wesley Publ. Co, Reading, Mass, 1983, reprint, Cambridge Univ, Press, Cambridge, 1967.
[12] R. Lidl, H. Niederreiter, Introduction to finite fields and their applications, Cambridge Univ, Press, Cambridge, second edition, 1994.
[13] R. G. Swan, Factorization of polynomials over finite fields, Pacific J. Math 12, 1962, 1099−1106

Received: November 27, 2013