Collision-resistant hash function based on two constraints

René Ndoundam [a,b], Juvet Karnel Sadié [b]

[a]UMI 209 IRD / UPMC UMMISCO, Bondy, France Project GRIMCAPE, LIRIMA
[b]Department of Computer Science, Faculty of Science, University of Yaoundé I, P.o. Box. 812 Yaoundé, Cameroon
E.mail : ndoundam@yahoo.com , karnel12@yahoo.fr

RÉSUMÉ. Une fonction de hachage cryptographique est une procédure déterministe qui compresse un ensemble de données numériques de taille arbitraire en une chaîne de bits de taille fixe. Il existe plusieurs fonctions de hachage : MD4, MD5, HAVAL, SHA... Il a été reporté que ces fonctions de hachage ne sont pas sécurisées. Notre travail a consisté à la construction d'une nouvelle fonction de hachage basée sur deux contraintes : la première vient des fonctions de hachage classique telles que MD4, MD5, SHA, HAVAL... et la deuxième est basée sur le théorème de Ryser (l'utilisation des tables de contingence de dimension 2).

ABSTRACT. A cryptographic hash function is a deterministic procedure that compresses an arbitrary block of numerical data and returns a fixed-size bit string. There exist many hash functions: MD4, MD5, HAVAL, SHA... It was reported that these hash functions are no longer secure. Our work is focused in the construction of a new hash function based on two constraints. The first constraint comes from the classical hash functions such as MD4, MD5, SHA, HAVAL... and the second one comes from the Ryser's Theorem (namely in the use of two-dimensional contingency tables).

MOTS-CLÉS : Matrices des zéros et des uns, fonction de hachage résistante aux collisions.

KEYWORDS : Matrix of zeros and ones, Collision-resistant hash function.

1. Introduction

A cryptographic hash function is a deterministic procedure that compresses an arbitrary block of data and returns a fixed-size bit string, the hash value (message digest or digest). An accidental or intentional change to the data will almost certainly change the hash value. Hash functions are used to protect the integrity of data or data signature.

There exists many hash functions : MD4, MD5, SHA-0, SHA-1, RIPEMD, HAVAL. It was reported that such widely hash functions are no longer secure [5]. Thus, new hash functions should be studied. Data security in two dimensional have been studied by many authors [2, 4]. In this paper, we propose a hash function based on the difficulty to solve a problem with two constraints than to solve a problem with only one constraint. The remainder of the paper is organized as follows. In the next section, we present some preliminaries. Section 3 is devoted to the design of hash function. Concluding remarks are stated in Section 4.

2. Preliminaries

For any integers a and p such that $0 \leq a \leq -1 + 2^p$, let us denote $bin(a,p)$ the decomposition of the integer a in base 2 on p positions. In other words :

$$bin(a,p) = x_{p-1}x_{p-2}\ldots x_1x_0 \quad \text{and} \quad \sum_{i=0}^{p-1} x_i \times 2^i = a$$

2.1. Two-dimensional

Let m and n be two positive integers, and let $R = (r_1, r_2, \ldots, r_m)$ and $S = (s_1, s_2, \ldots, s_n)$ be non negative integral vectors. Denote by $\mathfrak{A}(R,S)$ the set of all $m \times n$ matrices $A = (a_{ij})$ satisfying

$$a_{ij} = 0 \text{ or } 1 \text{ for } i = 1, 2, \ldots, m \text{ and } j = 1, 2, \ldots, n;$$

$$\sum_{j=1}^{n} a_{ij} = r_i \text{ for } i = 1, 2, \ldots, m;$$

$$\sum_{i=1}^{m} a_{ij} = s_j \text{ for } j = 1, 2, \ldots, n.$$

Thus a matrix of 0's and 1's belongs to $\mathfrak{A}(R,S)$ provided its row sum vector is R and its column sum vector is S. The set $\mathfrak{A}(R,S)$ was studied by many authors [1, 7]. Ryser [7] has defined an *interchange* to be a transformation which replaces the 2×2 submatrix :

$$B_0 = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

of a matrix A of 0's and 1's with the 2×2 submatrix

$$B_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$$

If the submatrix B_0 (or B_1) lies in rows k, l and columns u, v, then we call the interchange a $(k, l; u, v)$ *-interchange*. An interchange (or any finite sequences of interchanges) does not alter the row and column sum vectors of a matrix. Ryser has shown the following result.

Theorem 1 *[7] Let A and A^* be two m and n matrices composed of 0's and 1's, possessing equal row sum vectors and equal column sum vectors. Then A is transformable into A^* by a finite number of interchanges.*

Subsequently, for any $n \in \mathbb{N}$, we define the following functions :

$$f_0(n) = \lceil log_2(n+1) \rceil \qquad f_1(n) = 2n \times f_0(n)$$

$$f_2(n) = n^2$$

Let us consider a matrix $A \in \{0,1\}^{n \times n} \in \mathfrak{A}(R, S)$, i.e. its row sum vector R is such that $R \in \{0, 1, 2, \ldots, n\}^n$ and its column sum vector S is such that $S \in \{0, 1, 2, \ldots, n\}^n$. We define the function g_1 from $\{0,1\}^{n \times n}$ to $\{0,1\}^{f_1(n)}$ as follows :

$$\begin{aligned} g_1(n, A) = & bin(R(1), f_0(n)) || bin(R(2), f_0(n)) || \ldots || bin(R(n), f_0(n)) || \\ & bin(S(1), f_0(n)) || bin(S(2), f_0(n)) || \ldots || bin(S(n), f_0(n)) \end{aligned}$$

where $||$ denotes the concatenation. We note $|M|$ the length of the chain (or message) M. The size of A and $g_1(n, A)$ in terms of bits are respectively $f_2(n)$ and $f_1(n)$. It is easy to verify that g_1 is a compression function for $n \geq 7$.

Let us define :

– the function $VectMat$ which takes as input a vector $Vect$ of size n^2 and returns as output an equivalent matrix A of size $n \times n$.

– the function $MatVect$ which takes as input a square matrix A of order n and returns as output an equivalent vector $Vect$ of size n^2.

Let us consider a vector $x \in \{0,1\}^{p \times n^2}$, we define the function g_2 from $\{0,1\}^{p \times n^2}$ to $\{0,1\}^{f_1(n) \times p}$ as follows :

$$\begin{aligned} g_2(n, x) = & g_1(n, VectMat(x[1..n^2], n)) || g_1(n, VectMat(x[1 + n^2..2n^2], n)) || \ldots || \\ & g_1(n, VectMat(x[1 + (i \times n^2)..(i+1) \times n^2], n)) || \ldots || \\ & g_1(n, VectMat(x[1 + ((p-1) \times n^2)..p \times n^2], n)) \end{aligned}$$

where $x[i..j]$ denotes the concatenation of the elements at positions $i, i+1, i+2, \ldots, j-1, j$ of x, i.e.

$$x[i..j] = x[i] || x[i+1] || x[i+2] || \ldots || x[j-1] || x[j]$$

Comment : Let us consider two vectors C and D of size n^2 such that $g_2(n, C) = g_2(n, D)$, then by application of Theorem 1, we deduce that $VectMat(n, C)$ is transformable into $VectMat(n, D)$ by a finite number of interchanges. In fact, by definition, g_2 uses a concatenation of results from g_1. In this case, g_1(n,VectMat(n,C)) = g_1(n,VectMat(n,D)) and therefore C and D have equal row sum vectors and equal column sum vectors.
The conditions require to have a collision on g_2 on two inputs are not necessarily the same as for classical hash function : MD4, MD5, SHA-0, SHA-1, RIPEMD, HAVAL.

3. Design of hash function

3.1. Explanation of the idea

Example 1 :

In page 175 of paper [1], Brualdi gives the example of the following three matrices :

$$A_1 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \; ; \; A_2 = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix} \; ; \; A_3 = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

which belongs to $\mathfrak{A}(R, S)$ where $R = S = (2, 2, 1)$.
It is easy to verify that :

$$SHA1(MatVect(A_1, 3)) \; = SHA1(110110001) \; = \; 6e3da1d74147ca5d09413748e0bf6345c375af3e$$

$$SHA1(MatVect(A_2, 3)) \; = SHA1(110101010) \; = \; 7a481817a1014d06514de82e23c99d219e178b8a$$

$$SHA1(MatVect(A_3, 3)) \; = SHA1(110011100) \; = \; 4fbb6a6b6429262f8b62a93ed9b2f9b26bb7713d$$

It is easy to verify that :

$$\text{if } i \neq j \text{ then } SHA1(MatVect(A_i, 3)) \; \neq \; SHA1(MatVect(A_j, 3)) \qquad (1)$$

In his thesis Bart Van Rompay [6] presents some cases of attacks of the classical hash functions :

Attack of MD5 (see page 72 of [6])
Dobbertin [3] demonstrates that collisions are found on two messages blocks $\{W_j\}$ and $\{W_j^{'}\}$ $(0 \leq j \leq 15)$ with a small difference in only one of the words :

$$W_{14}^{'} = W_{14} + 1^{<<9} \qquad (2)$$

$$W_j^{'} = W_j \; (j \neq 14) \qquad (3)$$

Attack of HAVAL (see pages 76 and 77 of [6])
It is wrote in page 77 of [6], "we find such a collision for two messages blocks with a small difference in only one of the words :"

$$W_{28}^{'} = W_{28} + 1 \qquad (4)$$

$$W_j^{'} = W_j \; (j \neq 28) \qquad (5)$$

Collision are found on function g_2 if the two matrices have equal row sum vectors and equal column sum vectors. From the Example 1 above, we see that classical hash functions are not dependent of the theorem of Ryser.

Our design of a new hash function is based on the following facts :

– The condition defined by Ryser's Theorem is sufficient to attack the compression function g_2

– The condition defined by Ryser's Theorem is not sufficient to attack the classical hash function such as : MD5, SHA-0, SHA-1, RIPEMD, HAVAL, ...

3.2. Construction of a new hash function

Let us note H_1 a classical hash function such as : MD5, SHA-0, SHA-1, RIPEMD, HAVAL... From a hash function H_1, we build a new hash function H_2 as follows :

char $*H_2$ (int n, int Pos, File *x)
 1 : int $length$;
 2 : File * y ;
 3 : $length \leftarrow size(x)$
  // We pad x such that the size of y is the least multiple of n^2
 4 : $y \leftarrow x||1||0^d||bin(length, Pos)$
  // Pos represents the number of bits on which the length of x is
  // decomposed in base 2
 5 : return $H_1(g_2(n, y)||x)$
End

Remark 1 : The value of Pos depends on H_1 and n. n is a natural integer greatest or equal to 7. Let λ denote the maximum number of bits used in the representation of any input z of the function H_1. From the fact that for $n \geq 7$, g_2 is a compression function, we can define Pos as follows :

$$Pos = \lambda - 2.$$

3.3. Security of the function H_2

After the presentation of the hash function H_2, we now study in this subsection some attacks on H_2.

First Preimage attack :

Let us suppose that for an image y, we have find x such that $H_2(n, Pos, x) = y$, i.e.

$$H_1(z) = y \quad (6)$$

$$z = g_2(n, v)||x \quad (7)$$

$$v = x||1||0^d||bin(|x|, Pos) \quad (8)$$

The constraints defined by Equations (7) and (8) imply that First Preimage attack on H_2 is not weaker to solve than First Preimage attack on H_1.
Let us note $S1(n, Pos, y)$ and $S2(y)$ the sets defined as follows :

$$S1(n, Pos, y) = \{x|H_2(n, Pos, x) = y\}$$

$$S2(y) = \{x|H_1(x) = y\}$$

From the constraints defined by Equations (7) and (8), we deduce that it is possible that :

$$S2(y) \nsubseteq S1(n, Pos, y).$$

Second preimage attack :

We have an element x_1, we find x_2 such that :

$$H_2(n, Pos, x_1) \; = \; H_2(n, Pos, x_2) \tag{9}$$

To solve Equation (9), from an element w_1, we have to find w_2 such that

$$H_1(w_1) = H_1(w_2) \tag{10}$$

i.e., we have to find x_2 such that :

$$w_1 = g_2(n, y_1)||x_1 \tag{11}$$

$$y_1 = x_1||1||0^{d1}||bin(|x_1|, Pos) \tag{12}$$

$$w_2 = g_2(n, y_2)||x_2 \tag{13}$$

$$y_2 = x_2||1||0^{d2}||bin(|x_2|, Pos) \tag{14}$$

The constraints defined by Equations (11) and (13) imply that Second Preimage attack of H_2 is not weaker to solve than Second Preimage attack of H_1.
Let us note $S3(n, Pos, x_1)$ and $S4(x_1)$ the sets defined as follows :

$$S3(n, Pos, x_1) = \{x_2 | H_2(n, Pos, x_1) = H_2(n, Pos, x_2)\}$$

$$S4(x_1) = \{x_2 | H_1(x_1) = H_1(x_2)\}$$

From the constraints defined by Equations (11) and (13), we deduce that it is possible that :

$$S4(x_1) \not\subseteq \; S3(n, Pos, x_1)$$

Collision :

We want to find two elements x_1 and x_2 such that :

$$H_2(n, Pos, x_1) \; = \; H_2(n, Pos, x_2)$$

i.e. we have to solve the following problem : find x_1, x_2, z_1, z_2 such that :

$$H_1(z_1) = H_1(z_2) \tag{15}$$

$$z_1 = g_2(n, y_1)||x_1 \tag{16}$$

$$y_1 = x_1||1||0^{d1}||bin(|x_1|, Pos) \tag{17}$$

$$z_2 = g_2(n, y_2)||x_2 \tag{18}$$

$$y_2 = x_2||1||0^{d2}||bin(|x_2|, Pos) \tag{19}$$

The constraints defined by Equations (16) and (18) imply that Collision attack of H_2 is not weaker to solve than Collision attack of H_1.
Let us note $S5(n, Pos)$ and $S6$ the sets defined as follows :

$$S5(n, Pos) = \{(x_1, x_2) | H_2(n, Pos, x_1) = H_2(n, Pos, x_2)\}$$

$$S6 = \{(z_1, z_2) | H_1(z_1) = H_1(z_2)\}$$

From the constraints defined by Equations (16) and (18), we deduce that it is possible that :

$$S6 \nsubseteq S5(n, Pos).$$

Remark 2 : From the fact that $H_2(n, Pos, x)$ has the following form

$$\begin{aligned} H_2(n, Pos, x) = H_1(g_2(n, y)||x) \\ y = x||1||0^d||bin(|x|, Pos) \\ |y| \equiv 0 \pmod{n^2} \end{aligned} \tag{20}$$

Based on the three attacks and the above remark, we can easily deduce that any attack on H_2 is not weaker to solve than the same attack on H_1.

Example 2 :

Let us consider the two following texts x1 and x2 such that MD5(x1) = MD5(x2).

$$\begin{aligned} x1 = & d131dd02c5e6eec4693d9a0698aff95c \\ & 2fcab58712467eab4004583eb8fb7f89 \\ & 55ad340609f4b30283e488832571415a \\ & 085125e8f7cdc99fd91dbdf280373c5b \\ & d8823e3156348f5bae6dacd436c919c6 \\ & dd53e2b487da03fd02396306d248cda0 \\ & e99f33420f577ee8ce54b67080a80d1e \\ & c69821bcb6a8839396f9652b6ff72a70 \end{aligned}$$

$$\begin{aligned} x2 = & d131dd02c5e6eec4693d9a0698aff95c \\ & 2fcab50712467eab4004583eb8fb7f89 \\ & 55ad340609f4b30283e4888325f1415a \\ & 085125e8f7cdc99fd91dbd7280373c5b \\ & d8823e3156348f5bae6dacd436c919c6 \\ & dd53e23487da03fd02396306d248cda0 \\ & e99f33420f577ee8ce54b67080280d1e \\ & c69821bcb6a8839396f965ab6ff72a70 \end{aligned}$$

It is easy to verify that

$$MD5(x1) = MD5(x2) = EFE502F744768114B58C8523184841F3$$

By computation, we obtain :

$$H_2(16, 62, x1) \ = \ E8B4841671FF51D054071EB31BB03F1A$$

and

$$H_2(16, 62, x2) \ = \ 179CC5AFA8A2EA1BC0CC37CF2F9CFD3D$$

It is easy to see that : $H_2(16, x1) \neq H_2(16, x2)$ even when $MD5(x1) = MD5(x2)$.

Remark 3 : In the definition of the function H_2, if we replace the line 5 by the following

$$\text{return } H_1(g_2(n, y) \oplus x)$$

then we obtain another compression function. In this case, we can define Pos as follows :

$$Pos = \lambda.$$

4. Conclusion

From a hash function H_1, we have build a new hash function H_2 from which First Preimage attack, Second preimage attack and Collision are not weaker to solve than for the hash function H_1. This result is obtained by adding a new constraint in the resolution of the attacks. In general, solve a problem with two constraints is not weaker than solve the same problem with one constraint.

5. Bibliographie

[1] R. A. Brualdi, *Matrices of Zeros and Ones with Fixed Row and Column Sum Vectors*, Linear Algebra and its Applications, 33, 1980, pp. 159-231.

[2] L. Cox, *Suppression methodology and statistical disclosure control*, J. Amer. Statist. Assoc., 75(1980), pp. 377-385.

[3] H. Dobbertin, *The status of MD5 after a recent attack*, CryptoBytes, vol. 2, no2, pp. 1,3-6, 1996.

[4] I. P. Fellegi, *On the question of statistical confidentiallity*, J. Amer. Statist. Assoc., 67, (1972), pp. 7-18.

[5] Hongbo Yu, Xiaoyun Wang, *Multi-Collision Attack on the Compression Functions of MD4 and 3-Pass Haval*, Lecture Notes in Computer Science, 4817 , Springer 2007, pp. 206-226.

[6] B.V. Rompay, “Analysis and Design of Cryptographic Hash Functions, Mac Algorithms and Bloc Ciphers," (Doctoral dissertation, Katholieke Universiteit Leuven, 2004).

[7] H. J. Ryser, *Combinatorial properties of matrices of zeros and ones*, Canad. J. Math., Vol. 9, pp. 371-377, 1957.