Data Integrity Mechanism Using Hashing Verification

Rushdi A. Hamamreh
Computer Engineering Department
Faculty of Engineering
Al-Quds University

Mohammed A. Jamoos
Computer Science Department
Faculty of Science and Technology
Al-Quds University

Abstract
In this paper, we propose a new One-Way Hash Algorithm, which is also obtains better efficiency and security, compared with a particular conventional hash algorithm, Hash algorithm can be used to determine if two values are equal, a hash function maps keys to small values[1]. DILH algorithm using linear combination of matrices to find non-invertible matrix, that takes advantage about of the compact representation of a set of numbers in a matrix, and fast calculations.

IndexTerms
computational complexity, cryptography, hashing, collision, one-way algorithm, DILH, non-invertable matrix.

1. Introduction

There are three key aspects of data security: confidentiality, integrity and availability. In this paper, we introduced a concept of Data Integrity. One-Way Hash function have an important primitive cryptographic used for authentication, privacy and integrity[1][11].

Hash algorithm is a mathematical functions that converts a relatively large message into small strings of data and/or texts, to check whether the message has been trampled by attackers which could compromise the information, these strings of texts, called hashes, can be used [3][4].

More precisely, a hash algorithm $\mathbf{H}$ maps bits strings of arbitrary finite length to string of fixed length called hashes $\mathbf{H(x)}$ where $\mathbf{x}$ is the message block, given the original data, the encrypted data easy to calculate, for a given encrypted data to look for the original data is computationally infeasible, for different original data to get the same encrypted data is computationally infeasible [2][4][5].

Hash algorithm has played an important role in modern cryptography, it is generate because of the needs of message identify. So that it is widely used in the document verification and digital signature in information security[6]. According to its nature, you can't restore the original plaintext ($\mathbb{P}$) from the encrypted cipher text ($\mathbb{c}$), so it can't be used for conventional data encryption and it can only be used for identification[2].

MD5, SHA-1 are an excellent one-way hash algorithm. There are three main properties of such algorithms, in addition to one-way, it is collision-free and strongly collision-free [7].

Hash algorithm in practical use is mostly constituted of two parts, which could be called as compression function and iterative structure. So, we believe that the security of hash algorithm depends not only on the difficult problems, but also on the compression function structure and iterative of the hash algorithm [1][6].

There are many various hash algorithms including MD5, SHA-1, RIIPEMD and so on. However, MD5 has been broken by Wang Xiaoyun and she had found the effective method of attack the SHA-1 algorithm, so the difficulty of the break is reduced [8]. Thus, in order to meet practical application, it is desirable to study new hash generation algorithm .

In this paper, we describe non invertible matrix which can be used as multiplication matrix in Hill Cipher technique for one way hash algorithm. Hill cipher algorithm was used for symmetric encryption, where the multiplication matrix is the key. The Hill cipher requires the inverse of the matrix to recover the plaintext from cipher text. We propose a linear combination of matrix which is non invertible and easy to generate. We call this Data Integrity using Linear Combination for Hash Algorithm (DILH).

The rest of the paper is organized as follows. In section II, the matrix hash algorithm is introduced. In section III, the proposed DILH algorithm is presented. The analysis and results are shown in section IV. Conclusion remarks are drawn in section V.

The basic requirements for a cryptographic hash algorithm are:

- The input can be in any length.
- The output has a fixed length.
- $H(x)$ is relatively easy to compute for any given x .
- $H(x)$ is one-way.
- $H(x)$ is collision-free.

A hash function $\mathbf{H}$ is said to be one-way if it is hard to invert, where "hard to invert" means that given a hash value $\mathbf{h}$ [9],

Manuscript received September 5, 2014
Manuscript revised September 20, 2014

it is computationally infeasible to find some input x such that

$H(x) = V$.

For more secure transmitted data, padding using. Padding (Pad) is an algorithm where the plaintext is combined with a random values "rv" that is as long as the plaintext [1].

In our work , we have been used Byte padding. Byte padding can be applied for messages that can be encoded as an integral number of bytes.

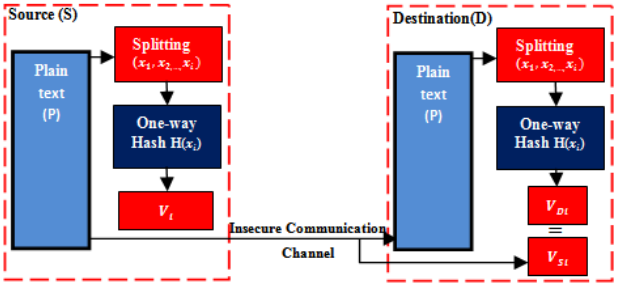

Figure.1 One-way hash algorithm

As it is shown in figure.1, if a given message x_1, it is computationally infeasible to find a message x_2 not equal to x_1 such that $H(x_1) = H(x_2)$ then H is said to be a weakly collision-free hash algorithm. On the other hand, H is a strongly collision-free hash algorithm if and only if it is computationally infeasible to find any two messages x_1 and x_2 such that $H(x_1) = H(x_2)$.

2. Matrix Hash Algorithm

Matrix K is a square matrix, the inverse is written K^{-1}. When K is multiplied by K^{-1} the result is the identity matrix I. Non-square matrices do not have inverses.

Not all square matrices have inverses. K square matrix which has an inverse is called invertible or nonsingular, and a square matrix without an inverse is called non-invertible or singular [12].

$$KK^{-1} = K^{-1}K = I$$

We call I here identity matrix. As we previously mentioned, not all matrices have Inverses but most of them have [12], so here we proposed an algorithm that converts any invertible matrix non-invertible one.

Hill Cipher

The core of Hill-cipher is matrix manipulations. It is a multi-letter cipher, is a type of monoalphabetic polygraphic substitution cipher. Hill cipher requires inverse of the key matrix while decryption. In fact that not all the matrices have an inverse and, therefore they will not be eligible as key matrices in the Hill cipher scheme [6][9]. Moreover, Hill cipher has several advantages such as disguising letter frequencies of the plaintext (P), its simplicity because of using matrix multiplication and inversion for enciphering and deciphering, its high speed, and high throughput .

A. Hill Cipher (Encryption , Decryption)

To Encrypt Plaintext Block size of x [9], we need key matrix ($K_{n \times n}$) with entries are between $(0, q-1)$ included, but the determinant must be relatively prime to p, each entry in the plaintext block is between $(0, q-1)$, included each block of plaintext is then an n-dimensional vector x .We encrypt vector x simply to produce the cipher text vector c using the following linear algebra equation

$c = x \times k \bmod m$

To Decrypt cipher text vector c [9], we need first to find the inverse matrix k^{-1} to k , where that matrix must be invertible over z_p , then can calculate x from the mathematical model

$$x = c \times k^{-1} \bmod m$$

In [5][9] they proposed new technique to convert any non-invertible matrix's to invertible ones. As a result, Hill cipher being a efficient algorithm because any encrypted text will decrypted using the key matrix [5][10].

B. Hill Cipher (Hashing algorithm)

The main point of one-way hash algorithm is that any encrypted text cannot be decrypted [11]. From this point ,we need to choose the non-invertible matrix from the hill cipher to use it inside the practical one- way hash algorithm.

$H(x) = x \times k \bmod m$

$H(x_i) = V_i$

Where k is the non-invertible matrix. In [11] author works on an algorithm that generate non-invertible matrix and multiply it by plaintext as column vector with modular value m to generate the hash value V.

3. Proposed Algorithm Models

Proposed algorithm DILH uses non-invertible matrix to produce hash value V. This algorithm selects and generates non-invertible matrixes using Linear combination[1] of rows or columns of a matrix to ensure that the hash values are collision-free and one-way properties.

In this section, we plot the diagram for our proposed algorithm which showing each step inside it and how it works , It is also showing the mathematical proof of our work. Besides that, we have the DILH algorithm analysis and result and its comparison with other hashing algorithms including SHA-1, MD5.

A. DILH algorithm

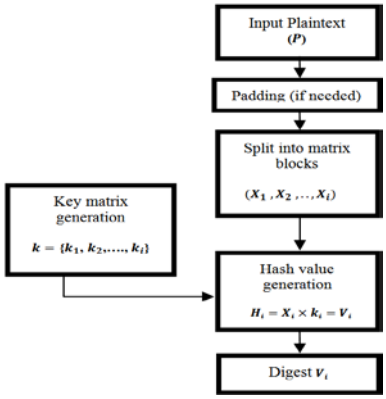

Figure.2 DILH diagram

According to figure 2, the step of DILH algorithm structured as the followings:

Step1 (Input): input $\mathbb{P}$.

Step2 (Padding): Pad .

Step3 (Splitting): $\mathbb{P}$ is split into . blocks $(X_1, X_2, \ldots, X_i)$, each of length $n \times n$ suitable for the hashing block.

Step4 (Key generation): Key matrix generation k_i :

4.1 Generate a random matrix $(\mathbb{R})$ with size $(n-1, n)$ and value in a interval (0,1).

4.2 Casting (R).

4.3 Introduce a new row in the matrix $\mathbb{R}$ so as the size of the resulting matrix k_i is $n \times n$. The added row is obtained by linear combination of row i and row j of matrix $\mathbb{R}$ according to:

$$K(N,:) = c_1 \times R(i,:) + c_2 \times R(j,:)$$

where c_1, c_2 are arbitrary integer constants. This ensure that the inverse of $\mathbb{K}$ denoted by $\mathbb{K}^{-1}$ does not exist.

Step5(Generation V_i) : Hash value V_i generation using this formula :

$$H_{xi} = X_i \times k_i = V_i$$

Step6(Digest) : Digest V_i.

Padding is the function where the plaintext is combined with a random data or "pad" to produce fixed length of the plaintext, and message digests are used to produce arbitrary values which is ensure data integrity [1].

B.Mathematical Model Proof

The first step is to define linear independence. Given a set of vectors[12]

$v_1, v_2, \ldots, v_k$

we look at their linear combinations

$c_1 v_1 + c_2 v_2 + \ldots\ldots + c_k v_k$

Where $c_1, c_2, \ldots, c_k$ are arbitrary constant weights. The trivial combination, with all weights $c_i = 0$, obviously produces the zero vector $0v_1 + \cdots + 0v_k = 0$. The question is whether this is the only way to produce zero. If so, the vectors are independent. If any other combination of the vectors gives zero, they are dependent [12].

It is well known that a matrix $\mathbb{A}$ is invertible if and only if its rows and columns are linearly independent [12]. This mean that if rows and columns of a matrix are linearly independent then the matrix is invertible, otherwise it's not invertible.

4. Analysis and Results

In this subsection we will analyze the time delay in the proposed DILH algorithm. Given a range of data files with sizes (8KB – 256KB), we first convert each file into matrix with size between (2x2 – 12x12). Then, we calculate the needed time in second to generate hash values. Table 1 shows the performance of the proposed DILH algorithm in terms of the required time to generate hashes for different file sizes and with different matrix sizes. According to Table 1, increasing the file size will increase the required time for all matrix size considered.

Table 1: Required time in seconds for the proposed DILH algorithm with different file size and matrix size m=127

File sizes in kilobyte (KB)

NxN	m=127					
	8	16	32	64	128	256
2x2	20.0618	39.9580	84.0139	211.8587	360.6685	679.7405
3x3	9.3918	24.701	36.9153	73.4639	146.4697	306.6645
4x4	9.1095	18.3049	36.8379	75.5187	152.6781	303.3392
5x5	5.8298	11.6831	23.5675	48.4917	98.0502	196.0664
6x6	5.8787	11.9484	23.9626	57.1672	112.0153	314.4286
7x7	6.0086	12.2762	24.7755	57.1958	111.8238	251.7975
8x8	6.0648	12.0529	23.9916	53.3537	107.8657	177.9706
9x9	5.9593	12.5579	23.6559	52.3380	103.0015	183.6245
10x10	4.7815	20.2727	23.4179	46.6217	98.0554	206.7475
11x11	5.5346	10.0950	21.8399	44.8996	88.1278	178.7949
12x12	5.5613	11.3943	22.7413	46.8754	93.1695	182.7774

Table 2: Required time in seconds for the proposed DILH algorithm with different file size and matrix size, m=256.

NxN	m=256					
	8	16	32	64	128	256
2x2	22.4047	47.5167	99.6309	209.8987	446.5694	851.0058
3x3	9.9071	19.7532	44.8870	90.7363	197.4561	338.1741
4x4	10.4507	22.9506	41.4800	92.2377	186.1238	355.4981
5x5	6.4367	13.7920	26.9642	60.0405	101.2355	214.9711
6x6	6.5498	14.1202	27.5373	63.6910	107.7837	233.1439
7x7	6.4847	13.8326	26.0235	48.7809	102.8205	228.2335
8x8	6.5144	12.4479	25.0766	46.6332	94.7677	224.3069
9x9	5.8912	12.0041	24.1447	43.8837	98.6085	212.9106
10x10	5.5588	10.8059	22.0636	42.6471	89.9993	178.3758
11x11	4.7090	10.7546	20.3674	38.6646	80.8281	175.8616
12x12	5.6618	10.9325	21.4436	40.8692	89.7404	174.0730

In addition, by Increasing the matrix size will usually decreasing the require time. Nevertheless, one can notice that there is an optimal matrix size that results in the lowest required time. For example, the optimal matrix size that

results in the lowest time when the file size 8KBand m=127 is 10x10, and when m=256 is 11x11. Table 2 summarizes the optimal matrix size for considered file sizes .

Table 3: Optimal time in seconds required for the proposed DILH algorithm.

File size/KB	Optimal matrix size $n \times n$		Time/Second	
	m=127	m=256	m=127	m=256
8	10x10	11x11	*4.7815*	*4.7090*
16	11x11	11x11	*10.0950*	*10.7546*
32	11x11	11x11	*21.8399*	*20.3674*
64	11x11	11x11	*44.8996*	*38.6646*
128	11x11	11x11	*88.1278*	*80.8281*
256	8x8	12x12	*177.9706*	*174.0730*

As shown in the previous table, the best time for calculating hash value at proposed model in 256KB is 8x8 when m=127 and 12x12 when m=256.

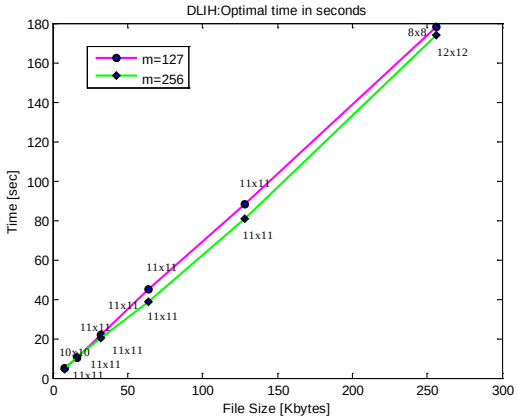

Figure 3: Optimal matrix size in terms of required time for different file sizes where m=127.

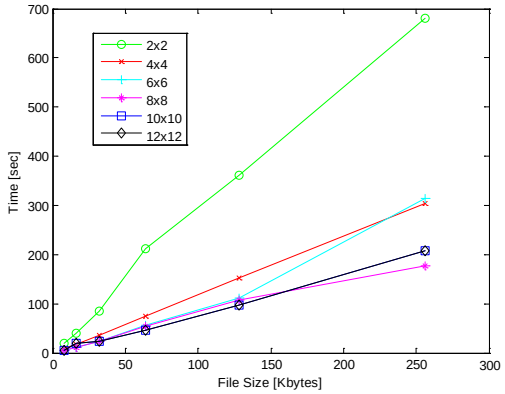

Figure 4: The time delay versus file size of the proposed algorithm for different matrix sizes and m=127.

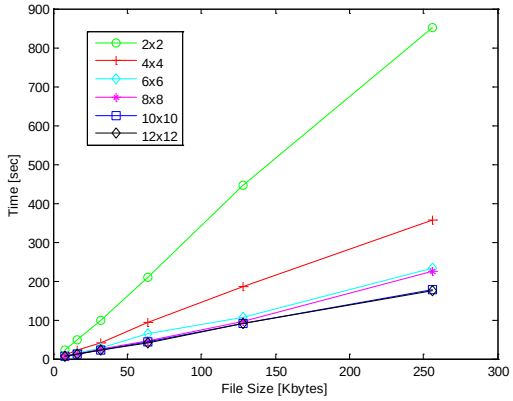

Figure 5: The time delay versus file size of the proposed algorithm for different matrix sizes and m=256.

According to the experimental data obtained by test, the efficiency of our hash algorithm is about 2.808 times slower than the efficiency of MD5 and about 5.813 times slower than SHA-1efficiency in 8KB file size. This ratio is increased in an positive relationship with file size, so you can find that our proposed algorithm is efficient 9.794 times that SHA-1 when the file size is 256KB. Table 3 summarizes the experimental delay time for considered hash algorithms .

Table 4: The time delay in seconds versus file size of the proposed algorithm compared to SHA-1 and md5 algorithm.

Data Size KB	Hash Algorithms			
	LDIH		SHA-1	MD5
	m=127	m=256		
8	*4.7815*	*4.7090*	*27.7990*	*13.2750*
16	*10.0950*	*10.7546*	*56.9980*	*25.3340*
32	*21.8399*	*20.3674*	*117.1002*	*51.3510*
64	*44.8996*	*38.6646*	*281.8830*	*135.4560*
128	*88.1278*	*80.8281*	*669.4310*	*432.0470*
256	*177.9706*	*174.0730*	*1743.1*	*1140.4*

From table 3, we can observe that our hash algorithm still has an advantage in the efficiency.

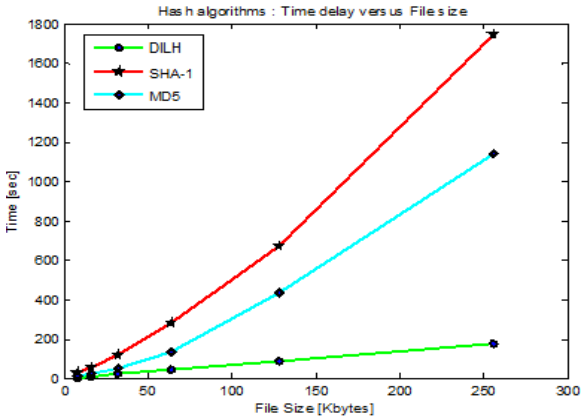

Figure 6: Hash algorithms comparison (SHA-1,MD5 and DILH) in term of time delay versus file size where m=127.

5. Conclusion and Future Work

The algorithm can protect user key, because one-way hash function is not reversible. We describe a new one-way data integrity hash algorithm based on special kinds of non-invertible metrics. Particularly, we proposed to generate non-invertible key matrices by linear combinations of rows and columns of a matrix. We have analyzed time delay of the proposed algorithm, compare it with other hash algorithms. In our experiment $m = 127,256$ and padding standard is ISO 10126.

We implement our hash scheme in a Dell vostro 1015 laptop, 2.10GHz 2 core(s), 2GB RAM/ Microsoft windows7/MATLAB 7.9.0 simulator.

From the result, our algorithm is faster than compared algorithm. Currently we are investigating collisions, hamming distance and vulnerable time for this algorithm. In addition, we will compare the proposed algorithm with other hashing algorithms including RIPEMD ,MD5 and SHA-1.

References

[1] William Stallings' , "Cryptography and Network Security Principles and Practices", 5th Edition, January 24, 2010.

[2] A New Hash Algorithm Based on MQ Problem and Polymorphic, Shangping Wang, Yaling Zhang, Youjiao Zou, Jin Sun, International Conference on Information Science and Technology,March 26-28, 2011 Nanjing, Jiangsu, China.

[3] Hash Function Vulnerability Index and Hash Chain Attacks. David Lee Department ofComputer Science and Engineering, The Ohio State University, Workshop on Secure Network Protocols, 2007. NPSec 2007. 3rd IEEE, 16-16 Oct. 2007.

[4] A one-time pad encryption algorithm based on oneway hash and conventional block cipher, Songsheng Tang, Fuqiang Liu, Qingdao University of Science and Technology, Qingdao, P.R.China, Consumer Electronics, Communications and Networks (CECNet), 2012 2nd International Conference on , 21-23 April 2012 .

[5] Invertible, Involutory and Permutation Matrix Generation Methods for Hill Cipher System, International Conference on Advanced Computer Control, National Institute of Technology Rourkela, Orissa-769008, India.

[6] A Class of Non Invertible Matrices in GF (2) for Practical One Way Hash Algorithm. International Journal of Computer Applications (0975 – 8887), Volume 54– No. 18, September 2012.

[7] Secure hash standard, federal information processing standards publication (Supersedes FIPS PUB 180 – 1993 May 11) .

[8] Wang X Y, Yu H B, Yin Y Q, Efficient collision search attacks on SHA-0,2005, Lecture Notes in Computer Science 3621 1.

[9] Rushdi A. Hamamreh, Mousa Farajallah(2009), "Design of a Robust Cryptosystem Algorithm for Non-Invertible Matrices Based on Hill Cipher", International Journal of Computer Science and Network Security; pp 11-16

[10] Data Encryption and Decryption by Using Hill Cipher Technique and Self Repetitive Matrix(2007). Amogh Mahapatra,Rajballav Dash.

[11] A Practical One Way Hash Algorithm based on Matrix Multiplication. Mohammed Abu Taha, Mousa Farajallah, Radwan Tahboub. s.l. : International Journal of Computer Applications, June, 2011, Vol. e 23-No.2. 0975-8887.

[12] Gilbert strang, "Linear Algebra and Its Applications", 4th Edition, May 9, 2011.

[13] Secure Hill Cipher Modifications and Key Exchange Protocol, Ahmed Y. Mahmoud and Alexander G. Chefranov. IEEE International Conference on Automation Quality and Testing Robotics (AQTR), 28-30 May 2010, volume 2 .

[14] DaniloGligoroski, Smile Markovski and Svein J. Knapskog(2006):" A Secure Hash Algorithm with only 8 Folded SHA-1 Steps" , IJCSNS International Journal of 194 Computer Science and Network Security, VOL.6 No.10.