TELKOMNIKA Telecommunication, Computing, Electronics and Control
Vol. 18, No. 5, October 2020, pp. 2543~2548
ISSN: 1693-6930, accredited First Grade by Kemenristekdikti, Decree No: 21/E/KPT/2018
DOI: 10.12928/TELKOMNIKA.v18i5.15276

A new algorithm for implementing message authentication and integrity in software implementations

Alaa Wagih Abdul Qader, Israa Ezzat Salem, Haider Rasheed Abdulshaheed
Baghdad College of Economic Sciences University, Iraq

Article Info

Article history:

Received Jan 11, 2020
Revised Mar 18, 2020
Accepted May 8, 2020

Keywords:

Authentication integrity
Hash functions
Validation

ABSTRACT

IT systems and data that you store, and process are valuable resources that need protection. Validation and reliability of information are essential in networks and computer systems. The communicating is done by two parties via an unsafe channel require a way to validate the data spent by one party as valid (or unaltered) by the other party. In our study, we suggest new one-way defragmentation algorithm to implement message authentication and integration in program execution. These software applications are readily available and freely available because most of the hash functions are faster than their existing radioactive blocks.

Corresponding Author:

Haider Rasheed Abdulshaheed
Baghdad College of Economic Sciences University,
Baghdad, Iraq.
Email: haider252004@yahoo.com

1. INTRODUCTION

Authentication and message integrity are done by several methods. Parallel encryption mechanisms can be used but have their disadvantages. Tsudik highlighted the [1] disadvantages such as speed, cost factor, optimization of data sizes and so on. These methods combine the functions of confidentiality and authentication. But there are scenarios where the complete message is not required. Such as applications, messages storing confidential is not a concern, but authentication is important. For example, in SNMP it is usually important for a managed system to authenticate incoming SNMP commands (such as alteration parameters in the management system) but hiding SNMP traffic is not required. To implement message authentication, alternative techniques (except that mentioned in the last paragraph) are fragmentation functions or MAC. Mac system makes long serial zeros such as DES. Idea of creating MAC devices is started spreading from cryptographic fragmentation functions [2]. See Figure 1.

2. CRYPTOGRAPHY

The encryption could provide many roles in the authentication. The authentication capabilities use of common cryptographic keys, which are only owned by licensed entities. The encoding plays an important role in authentication and identification in two ways, wherever the encoding support great role of the authentication. The encryption can give security for data transformation and storing authentication data. Furthermore, the encryption could use it as an authentication method [3, 4]. The encryption supports authentication by using it in the authentication systems, ex. systems of the password encryption are usually

used encrypted password and use code encryption and card system to protect important information, also use password creator for random encryption passwords. The encryption is usually used in particular applications to transfer authentication information and data from one to another system by a network as well as authentication systems by encryption depend on private cryptosystems or public key cryptosystems. The main cryptographic systems use one key for decryption and encryption functions. Authentication systems based on private cryptographic systems rely on a common key between the user who logs in and the authentication system. Common cryptographic systems decode encrypting and decrypting functions by separate among the keys. Encrypted authentications systems depend on the public key of the cryptographic systems rely [5-7].

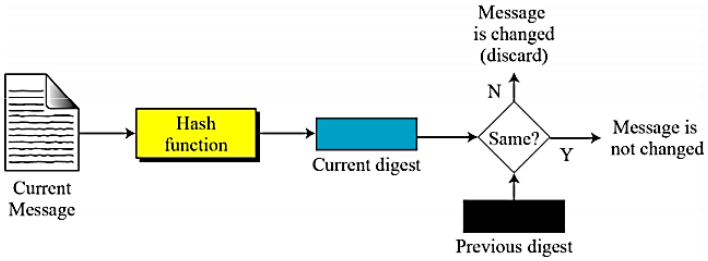

Figure 1. Integrity of message [2]

3. HASH FUNCTION

The Hash algorithm tracks binary data, and produces an intensive representation, called the "Message Summary". The encryption algorithm is a Hash algorithm made for security features. The Standard HASH-180 including (SHA-256, SHA-1, SHA-224, SHA-384, and SHA-512, for more information the standard hash is common used by modern information technology. In last years, there are several attacks against NIST algorithms, but it failed, especially against SHA-1. NIST held many public workshops for estimation the status of algorithms. Depend on the workshops; NIST suggested more than one additional algorithm by generic competition. And NIST suggested a timetable for testing and published new using policy of the Hash functions [8, 9].

4. CRYPTOGRAPHIC HASH FUNCTION

The function of hashing fragmentation is a necessary procedure takes an blocking of the arbitrary data and re-backs a fixed size bit string, and coding value of the hash, like changes of hash values. The encoded data are usually being the message, often named the message segmentation value or summarized merely. The cryptographic function has several characteristics:

- Easy to calculate the retail value of a particular message
- Not feasible for looking for a message containing a specific segmentation
- Useless to changing message without any modified the tick
- Not feasible to look for two messages in the same ticker

Encryption functions contain several data security software, particularly in message authentication and digital signatures. It can also use normal retail functions, for index data in the retail tables, to take fingerprints, to determine repeated same data or unique data, and as a checksum to determine any corruption in the data [10, 11]. The function of the HASH encryption should be including all kinds of cryptanalytic attack. That should have the following characters:

- Preimage resistance: Because of fragmentation, it's very hard to found a message. That linked to the concept of the one-way function. Without that are vulnerable.
- Second preimage resistance: It's included the inputs wherever finding other inputs. Sometimes that indicates for low impact resistance, without these functions are mean previous pre-attack attacks [12].

4.1. Cryptographic hash functions security

4.1.1. Verifying messages integrity

Message integrity is one of the applications for secure fragmentation. By that could determine the changes to a message, are it has been made, especially in the comparing between before and after sending the message. Most digital signature algorithms are used to confirming the validity of the signed message,

wherever the signature is proof of message authenticity. The particular and specific application is doing by password confirmation. Wherever the passwords are stores as summary text. For authenticate of the user, the password is divided and compared to the stored test.

4.1.2. File or data identifier

Often coding management systems, example Mercurial, Git, Monotone and sha1sum use many kinds of content (pedigree information, directory trees and file content, etc.) to make it unique. The function of the encoding is determining file-sharing networks (peer-to-peer), ex. in an (ed2k) link, an (MD4) variable is merged with the file size, wherever, giving sufficient data to identification files sources and download the file. Magnet bonds are another example. This fragmentation is often the highest segmentation in the retail list or the retail tree that allows other benefits. The function of the hash applications allows a quick search of data in a hash table. A specific type of Hash functions, the encryption fragmentation functionality offers itself well for this application as well. However, compared to standard retail functions, functions of the encryption defragmentation are more expensive than the calculation wherever it is necessary to protect against the fraud creating information abstracted from expected data [13-15].

4.2. Hash functions depend on block ciphers

Many methods are used for block cipher for build function of the cryptographic hash, particularly function of the one-way compression. The methods same block cipher methods, wherever both used for encryption. The hash functions are example MD5, MD4, SHA-2 and SHA-1 and are designed for the same purpose, SHA-3 finalists have functions with block-cipher such as (e.g. BLAKE, Skein,) and (e.g., Keccak, JH) [16-18]. A standard block cipher like (AES) used instead of block ciphers. They use blocks and large keys, could changing all keys of all block, and make to resistance key attacks. There are several goals of the general ciphers. AES has key sizes which make it nontrivial to using to create long hash values; when key changes each block, the AES encryption becomes without or less efficient; and key attacks make it without security [19]. Figure 2 shows the Merkle–Damgård construction.

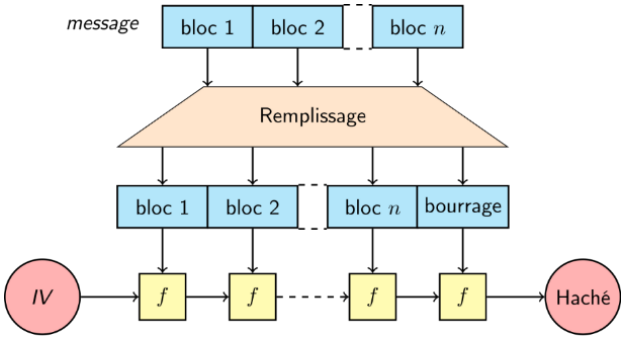

Figure 2. Merkle–Damgård construction [19]

4.3. Cryptographic encryption algorithms

The hash function used in computer sciences, and it means string compression function of arbitrary input to a series of fixed length. But, if it has some of the additional requirements, it could use it in the encryption applications. There are many encoded functions, but many of it is weak. If the jamming function is not a success, a successful attack against it decreases expert confidence. Ex. in 2004, vulnerabilities were occurring in a hash function such as RIPEMD, MD5 and SHA-0. That will be provide long-term security of the subsequent function derived from these HAASH example, SHA-1, RIPEMD-128 and RIPEMD-160. RIPEMD and SHA-0 are not commonly using and it replaced by a new improved version. MD5 and SHA-1 daltons become commonly used at 2009, although all that MD5 was broken wherever the attack was done against it in 2008. They are developing SHA-0 and SHA-1 function by NSA. Also, a strong attack was done on SHA-1 in 2005. Furthermore, successful, strong attack occurred in 2005 on SHA-1 and found collisions in (263) operations. Also, there are weaknesses points of SHA-1, suggesting that it need several years to break it. There are other new applications protect against the matters by using another SHA, like SHA-2, or new techniques like random defragmentation [20-23]. To produce new applications, use Hash functions, there is strong competition for developing SHA-2 to produce SHA-3 in 2012 [24, 25]. The following algorithms are usually used in cryptography as Figure 3.

Comparison of SHA functions

Algorithm and variant		Output size (bits)	Internal state size (bits)	Block size (bits)	Max message size (bits)	Rounds	Operations	Security bits (Info)	Capacity against length extension attacks	Performance on Skylake (median cpb)[1] long messages	Performance on Skylake (median cpb)[1] 8 bytes	First Published
MD5 (as reference)		128	128 (4 × 32)	512	Unlimited[2]	64	And, Xor, Rot, Add (mod 2^{32}), Or	<64 (collisions found)	0	4.99	55.00	1992
SHA-0		160	160 (5 × 32)	512	$2^{64} - 1$	80	And, Xor, Rot, Add (mod 2^{32}), Or	<34 (collisions found)	0	≈ SHA-1	≈ SHA-1	1993
SHA-1								<63 (collisions found[3])		3.47	52.00	1995
SHA-2	*SHA-224*	224	256 (8 × 32)	512	$2^{64} - 1$	64	And, Xor, Rot, Add (mod 2^{32}), Or, Shr	112	32	7.62	84.50	2004
	SHA-256	256						128	0	7.63	85.25	2001
	SHA-384	384	512 (8 × 64)	1024	$2^{128} - 1$	80	And, Xor, Rot, Add (mod 2^{64}), Or, Shr	192	128 (≤ 384)	5.12	135.75	
	SHA-512	512						256	0	5.06	135.50	
	SHA-512/224	224						112	288	≈ SHA-384	≈ SHA-384	
	SHA-512/256	256						128	256			
SHA-3	*SHA3-224*	224	1600 (5 × 5 × 64)	1152	Unlimited[4]	24[5]	And, Xor, Rot, Not	112	448	8.12	154.25	2015
	SHA3-256	256		1088				128	512	8.59	155.50	
	SHA3-384	384		832				192	768	11.06	164.00	
	SHA3-512	512		576				256	1024	15.88	164.00	
	SHAKE128	*d* (arbitrary)		1344				min(*d*/2, 128)	256	7.08	155.25	
	SHAKE256	*d* (arbitrary)		1088				min(*d*/2, 256)	512	8.59	155.50	

Figure 3. Family of hash functions [17]

4.4. A proposed hash algorithm

A hash function should have the ability to processing random length message into an output have fixed length. Also, possible breaking fixed length input into a series of blocks has equal size and using a one-way compression function. In this algorithm, we assumed three rounds, in each round there is 32-iteration. The output of any length of the input is 96-bit. We used logical operators (and, or, xor, shl, shlr, shrr).

4.4.1. Outline of a proposed hash algorithm

The first step is generated of 64 key [311...2] plus 32-bit proposed key). Secondly, it takes an input message and breaks into n-512-bit blocks then each block break into of 64 word (32-bit) modified using (special equations to re-change original word), see Figure 4. Then take each block and do two special rounds on it; finally do special equations on A, B, C, D (32- bit variables), see figure and then contraction of them, see Figure 5.

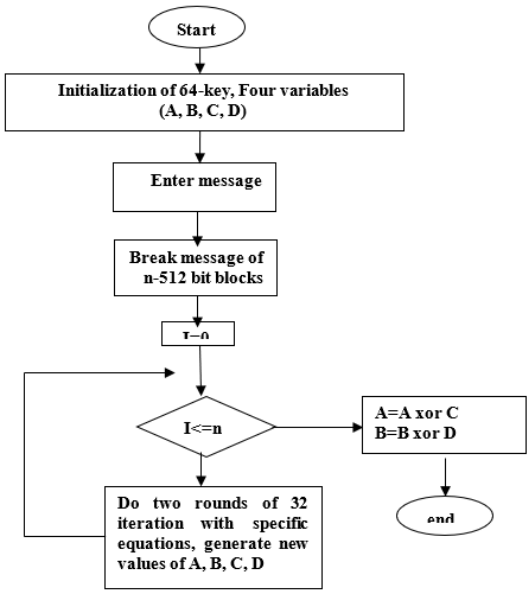

Figure 4. Outline of new one-way hash function flowchart

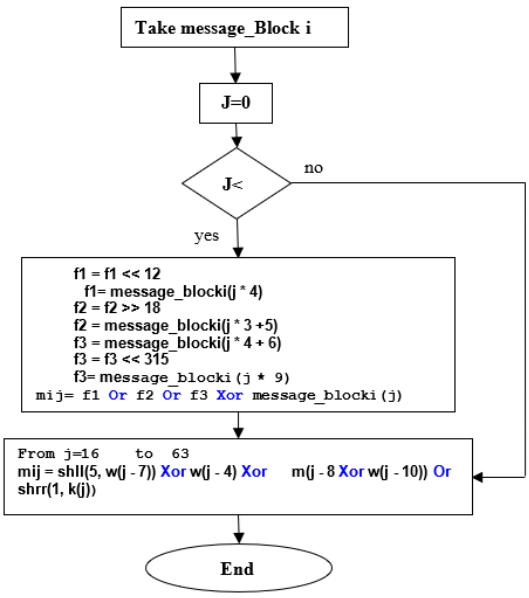

Figure 5. Outline the flowchart to re-change the original word

4.4.2. The first proposed function:

In this function ,we shift left the first variable A first by 9 and secondly shift right by 7 the exclusive or of them ;also shifting the variables (B by left 3, C by right 4 and D by left 5) then exclusive or of them ;because the best shifting number that its obtained exactly different results also not used in another hash functions after examined different numbers of shift and different another logical operators, see Figure 6.

4.4.3. The second function

In this function, we shift left rotate the first variable A first by 12 and secondly shift right rotate by 15 the exclusive or of them. Also, exclusive or of not variable C and variable D, because the best shifting number that its obtained exactly different results also not used in another hash functions after examined different numbers of shift and different another logical operators, see Figure 7.

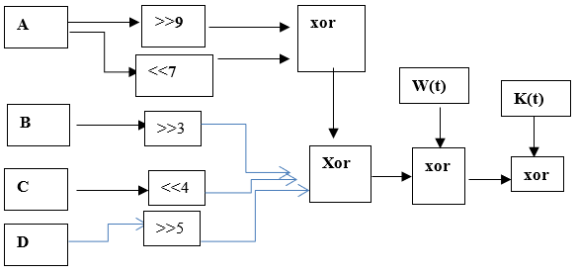

Figure 6. Specific equation for the first round

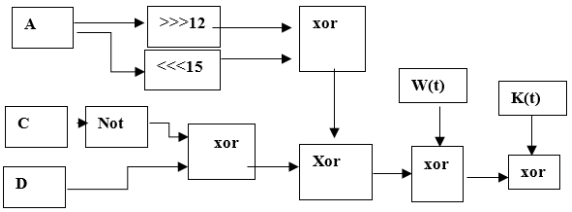

Figure 7. Specific equation for the second round

4.4.4. Variables modification

To make a perfect modification after TWO rounds variables will be modified by special formal to enhance the results with exactly different others, it make use of Xor ,shl, shrr, see Figure 8. For the Complexity measurements. The hash function is looking for message agreed with a given message by using a brute force search 2^L evaluations, where L represents bits number of the key with proposed variables length. That named pre-image is attacking. And the second factor is looking for two messages (different) which produce the same message digest, that named collision, and that need to average only $2^{L/2}$ evaluations using a birthday attack. In the proposed hash algorithm, we used 32-bit key length in addition to the proposed three variables each of 32-bit length; so, the brute force will make an exhaustive search of degree 2^{128}.

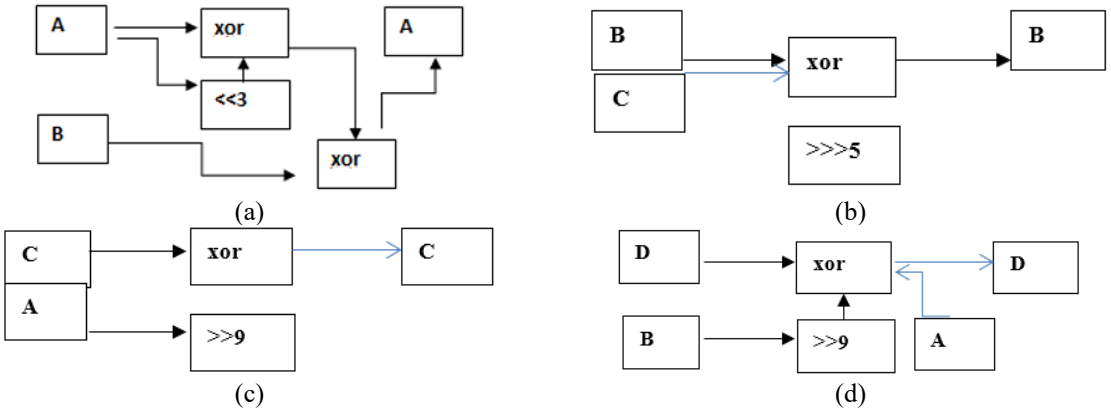

Figure 8. Update of variables; (a) variable A, (b) variable B, (c) variable C, (d) variable D

5. CONCLUSIONS

We have shown in this paper how cryptographic hash functions cannot find two different messages have resembled hash value. It is not broken because of the degree of confusion and complexity is very high, and an attacker cannot fake a signature in a particular computing environment. any message will be saved and will get sign, if the messege is changed will get new signature.

REFERENCE

[1] S. B. Thigale, R. K. Pandey, P. R. Gadekar, and V. A. Dhotre, "Lightweight novel trust-based framework for IoT enabled wireless network communications," *Periodicals of Engineering and Natural Science*, vol. 7, no. 3, pp. 1126–1137, 2019.

[2] M. Zia and R. Ali, "Cryptanalysis and improvement of an elliptic curve based signcryption scheme for firewalls," *PloS One*, vol. 13, no. 12, pp. e0208857-e0208857, 2018.

[3] S. Dey, S. Sampalli and Q. Ye, "MDA: message digest-based authentication for mobile cloud computing," *Journal of Cloud Computing*, vol. 5, no. 1, pp. 1-13, 2016.

[4] M. Harran, W. Farrelly and K. Curran, "A method for verifying integrity & authenticating digital media," *Applied Computing and Informatics*, vol. 14, no. 2, pp. 145-158, 2018.

[5] L. Ferretti et al., "A symmetric cryptographic scheme for data integrity verification in cloud databases," *Information Sciences*, vol. 422, pp. 497-515, 2018.

[6] M. A. Alazzawi et al., "Efficient Conditional Anonymity with Message Integrity and Authentication in a Vehicular Ad-Hoc Network," *IEEE Access*, vol. 7, pp. 71424-71435, 2019.

[7] K. Ding, S. Chen and F. Meng, "A Novel Perceptual Hash Algorithm for Multispectral Image Authentication," *Algorithms*, vol. 11, no. 1-14, pp. 6, 2018.

[8] B. Alomair, "Authenticated encryption: how reordering can impact performance," *Security and Communication Networks*, vol. 9, no. 18, pp. 6173-6188, 2016.

[9] P. Vijayakumar et al., "Computationally efficient privacy preserving authentication and key distribution techniques for vehicular ad hoc networks," *Cluster Computing,* vol. 20, no. 3, pp. 2439-2450, 2017.

[10] S. Mahjabin, "Implementation of DoS and DDoS attacks on cloud servers," *Periodicals of Engineering and Natural Science*, vol. 6, no. 2, pp. 148–158, 2018.

[11] N. Ghose, "Authentication and Message Integrity Verification without Secrets," Dissertations, The University of Arizona, 2019.

[12] J. Jenefa and E. A. Mary Anita, "An Enhanced Secure Authentication Scheme for Vehicular Ad Hoc Networks Without Pairings," *Wireless Personal Communications,* vol. 106, no. 2, pp. 535-554, 2019.

[13] C. K. Lim et al, "Design and development of message authentication process for telemedicine application," *2018 IEEE Conference on Wireless Sensors (ICWiSe),* 2018. doi: 10.1109/ICWISE.2018.8633289.

[14] P. Mundhenk et al, "Security in Automotive Networks: Lightweight Authentication and Authorization," *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, vol. 22, no. 2, pp. 1-27, 2017.

[15] H. Li et al, "Cumulative Message Authentication Codes for Resource-Constrained Networks," arXiv.org, 2020.

[16] G. K. Sodhi et al., "Preserving Authenticity and Integrity of Distributed Networks through Novel Message Authentication Code," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 12, no. 3, pp. 1297-1304, 2018.

[17] S. M. S. Hussain, S. M. Farooq, and T. S. Ustun, "Analysis and Implementation of Message Authentication Code (MAC) Algorithms for GOOSE Message Security," *IEEE Access*, vol. 7, pp. 80980–80984, 2019.

[18] J. W. Seo and S. J. Lee, "A study on the integrity and authentication of weather observation data using Identity Based Encryption," *Springerplus*, vol. 5, no. 1, pp. 1, 2016.

[19] N. Venkateswaran, A. Shekhar, and S. Changder, "Using machine learning for intelligent shard sizing on the cloud," vol. 7, no. 1, pp. 109–124, 2019.

[20] J. Noh, S. Jeon, and S. Cho, "Distributed Blockchain-Based Message Authentication Scheme for Connected Vehicles," Electronics, vol. 9, no. 1, p. 74, Jan. 2020.

[21] S. Rashid, A. Ahmed, I. Al Barazanchi, and Z. A. Jaaz, "Clustering algorithms subjected to K-mean and gaussian mixture model on multidimensional data set," Period. Eng. Nat. Sci., vol. 7, no. 2, pp. 448–457, 2019.

[22] C. Biswas, U. Das Gupta, and M. M. Haque, "An Efficient Algorithm for Confidentiality, Integrity and Authentication Using Hybrid Cryptography and Steganography," in 2nd International Conference on Electrical, Computer and Communication Engineering, ECCE 2019, 2019.

[23] D. Engels, M. J. O. Saarinen, P. Schweitzer, and E. M. Smith, "The hummingbird-2 lightweight authenticated encryption algorithm," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2012.

[24] A. J. Menezes, P. C. van Oorschot, S. A. Vanstone, A. J. Menezes, P. C. van Oorschot, and S. A. Vanstone, "Hash Functions and Data Integrity," in Handbook of Applied Cryptography, 2018.

[25] K. Mahmood, S. Ashraf Chaudhry, H. Naqvi, T. Shon, and H. Farooq Ahmad, "A lightweight message authentication scheme for Smart Grid communications in power sector," *Comput. Electr. Eng.,* vol. 52, pp. 112-24, 2016.