Mathematical Message Authentication Code Using S-Box key

Mohammed Ali Mohammed[†] , Loay K. Abood[†] and Makki Maliki[†]

[†]University of Baghdad, College of Science, Baghdad, Iraq

Summary
In recent year the exchange of information is become widely used. This information is accessible to all, so security is one of the important concept should provide for this information. Message Authentication is one of concept in digital world that is used to verify the integrity of message, message authentication confirm that the data received and send are same (means no modification, deletion or insertion). This paper aims to design and implementation new method for calculation Message Authentication Code (MAC) based on shared secret substitution-box (S-Box) key to provide the message integrity. The methodology consists of four steps: firstly generation the S-Box table using the shared secret key. Secondly, split the message M into blocks (each block will contain number of bits). Thirdly, apply the substitution operation for each character in each block (depend on the index of character in block and block number) using S-Box table. Finally, use mathematical operation to generate the MAC from blocks. The MAC will attach to message and send to receiver, the receiver compare the incoming MAC with calculator MAC, if there is a mismatch the receiver knows that the message has been modify. The result shows, without any doubt, are faster (no rounds), more secure, and less complex comparing with standard method of MAC.

Key words:
Message Authentication Code (MAC); Message Authentication; Message Integrity; Substitution-Box (S-Box).

1. Introduction

Data security is an interesting topic as a fastest evolution of data communication over unsecured channel. Data in digital world can be easily captured by hackers. The problem statement in data security is that the message can be modified during transmit without been distinguishing by the receiver.

Message Authentication Code (MAC) is a keyed hash function, the input message can be any variable length, but the output tag has to be fixed length. The MAC means that the message should arrived exactly as same as sender sends. The generated tag is used to determine that how the MAC algorithm is protected [1]. There are three actors of the standard message authentication. These actors are sender, receiver and attacker. The sender and the receiver share secret key(s), while the attacker tries to have this share key(s) by doing brute force attack, or corrupt the original message.

The main characteristics of our algorithm are using simple mathematical operation when apply summation operation to the 3 number. Moreover, the generated tag from the substitution message not from the original message to avoid the cryptanalysis attack. For more complexity the substitution operation is indirect as the operation is not only based on the S-Box but rather is based on the following: the index of character, block number, and weight of character. The reset of the paper is organized as follows: in section 2 we discuss the related work of MAC algorithm, while section 3 deals with design and generate the S-Box table. In the section 4 we describe our methodology to generate the tag. The requirements and properties of our methodology shows in section 5 .In section 6 we show the experimental results and discuss the result. Finally, we conclude this paper in section 7.

2. Related Work

There are several ways to generate MAC that are different in complexity, time execution and security.

Petrank and Rackoff [2] presented Encryption the CBC (Chaining Block Cipher) - MAC (EMAC) value with new key. They proved that EMAC is secure when the message length is a positive multiple of the block size. In William Stalling [3] proposed a keyed-hash message authentication code (HMAC). This method becomes widely used as MAC function and spread implemented in many applications and protocols. The weakness of this algorithm is that the executions time is twice long comparing with hash function. The main reason behind that is the HMAC includes two executions of the hidden hash function for every output block.

Keting Jia, et al. [4] tested the CBC-MAC without the truncation function and showed that the CBC-MAC is not secure when message is variable length. The authors provided a prove that the Message Authentication Code (MACs) such as CBC-MAC, three-key encipher CBC mode, OMAC [5], PC-MAC, TMAC, FCBC, XCBC, EMAC, ECBC, MT-MAC, CMAC are vulnerable to second pre-image attack.

Building two types of CBC-MAC, called GCBC1 and GCBC2, which presented by the Mridul Nandi [6]. This algorithm is been proven that the length extension attack

Manuscript received July 5, 2017
Manuscript revised July 20, 2017

could be happened when the message input is variable length, in this state the CBC-MAC is insecure.

Neeta wadhwa, et al. [1] designed a single method to generate the integrity, authentication, confidentiality and availability. This method does not generate a fix length of MAC for all variable length of a message as well. It generate quarter of the length of original message. While the standard properties of MAC algorithm obtain to generate a fix output (tag) length.

Nicky Mouha, et al. [7] presented multi MAC algorithms, which are the Universal MAC [8], GMAC [9], and Poly-1305-AES [10]. These methods are based on universal hash functions that used a nonce number as input. The nonce number must use for one time only to avoid the forgery attack. The researchers concluded when the tags are truncated the GMAC and Poly-1305 will become insecure.

3. Design and Generate S-Box Key

Substitution-Box (S-Box) is a non-linear transformation and implemented as lookup table which makes the correlations between the key and the cipher-text as complex (confusion). The substitution-box is widely used in most new encryption algorithm. It is used to prove the robust cryptographic primitive versus differential cryptanalysis and linear transformation [11]. S-Box is one of major properties that is used to determine the strong of symmetric cryptographic. The purpose of the S-Box table is to replace each input byte with another output byte by substitution operation. There are two types of S-Box which are static and dynamic. In static S-Box there is no relation with the cipher key as well as contains of S-Box are not depend on the secret key. Therefore, the secret key is used only to change the address of S-Box. Thus, the static S-Box will produce same output in each round. So, the static S-Box is easy to cryptanalysis. The dynamic S-Box has a relation with the secret key and the output is changing in each round according to a secret key. Therefore, this type is more secure than the static. The dynamic S-Box can be generated by Pseudo Random Sequence Generator algorithm [12].

In this paper, the generation of S-Box table depends on the two important values. Firstly, the shared secret key and secondly the Pseudo-Random Number Generators (PRNG) which depends on the four prime numbers. The S-Box structure contains 16 rows and 256 columns (16x256). The 16 rows represent the block size of message (16-byte) and the 256 columns represent the range of ASCII characters (Contain the printable and non-printable characters).

3.1 S-Box Construction [13]

S-Box is constructed by 4096 elements (16x256) using Pseudo-Random Number Generation algorithm to generate range numbers between [0…256] and generate positions in range of [0… 4095]. Then each row in S-Box will circularly shift left/right according to value of the secret key. The complete algorithm (algorithm 1) is shown below:

- **Pseudo-code to Generate Pseudo-Random Numbers:**
  a. Initial two large prime numbers x , y
  b. For loop i=0,1,2,…, 4095 do
  c. x=(x*y+1) mod 256
  d. array_of_numbers(i) = x
  e. end for loop
- **Pseudo-code to Generate Random Numbers for positions**:
  a. Initial two large prime numbers a , b
  b. For loop j=0,1,2,…, 4095
  c. a=(a*b+1) mod 4096
  d. array_of_positions(j) = a
  e. end of loop
- **Generate S-Box table**:
  a. Initial key as vector of integer, array_of_numbers, array_of_positions, rounds
  b. Fill S-Box(I , j) from array_of_numbers, where i=0,…,15 and j=0,…,255
  c. For loop round=1,…,rounds do
  d. For loop row=0,…,15 do
  e. If key(row) is even then row in S-Box is circularly shifted to left of key(row).
  f. If key(row) is odd then row in S-Box is circularly shifted to right of key(row).
  g. End for row loop
  h. End for round loop
  i. All element in S-Box are permuted according to values in array_of_positions
  j. Now the S-Box is generated.

After generate the S-Box, as shown in Fig 1, we conclude that the robust S-Box does not depend on the generation method. To prove this conclusion, the result is compared with security parameters such as balanced output with 0(s) and 1(s), Avalanche effect, Hamming distance between

words, and finally less time execution compare with exist algorithm.

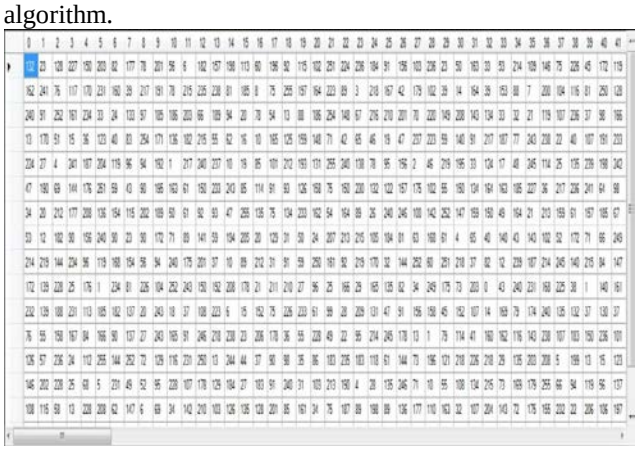

Fig. 1 Substitution-Box Data.

One of the most known attack algorithm is by count each character frequency which based on Language properties, such algorithm is called cryptanalysis attack. Therefore, to prevent this type of attack, the S-Box algorithm generate similar character`s frequency, as shown in Fig 2.

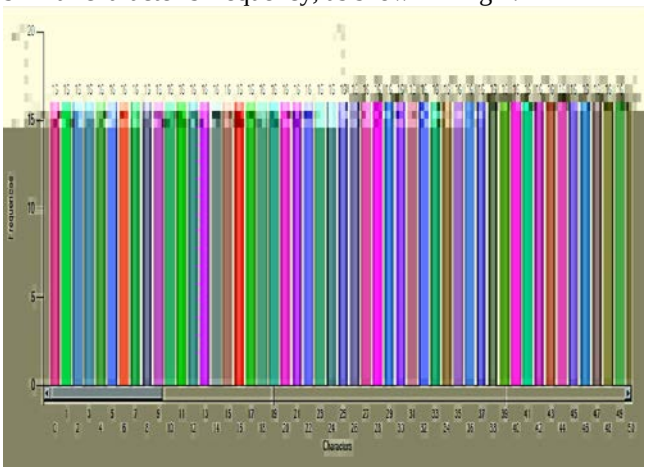

Fig. 2 Substitution-Box Histogram.

4. Methodology

New design and implementation of proposed algorithm (Mathematical Message Authentication Code (MMAC)) used to verify the integrity of a message. The proposed algorithm produces MAC (tag) for the message. This message then will be sent including its MAC. In the receiver side, calculate the new MAC for this message to compare with the received one for matching propose. Fig. 3 show this process.

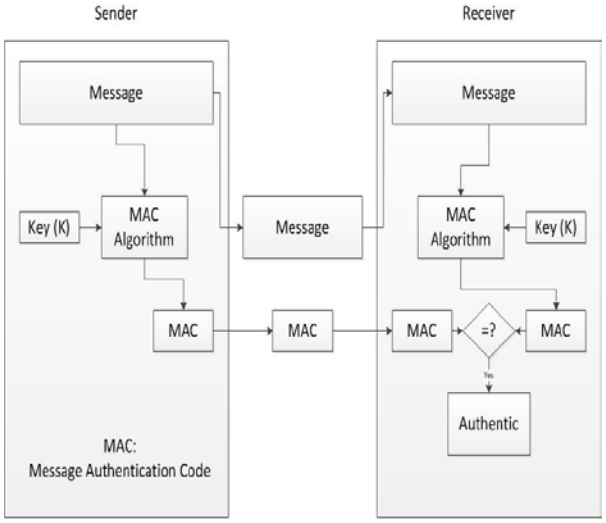

Fig. 3 Shows the sender and receiver process

To construct a MAC, there are three common algorithms: construction based on universal hash, block cipher, and cryptographic hash function. In this paper, our MAC construction is based on block cipher algorithm [4].

Algorithm (2): MAC Construction Algorithm:

- **First Step (Generate the S-Box):** in this step the proposed algorithm generates the S-Box table (as Sub-Key Generation) by using: the above proposed method (Algorithm 1), shared secret key, and four prime numbers. The generated S-Box table will use in third step.

- **Second Step (Preprocessing):** the preprocessing step contain padding bytes (key characters) to end of message to complete the final block size if it is not completed. Then split the input message (M) to the blocks according to n-bytes. The block size should equal or larger than 128-bits to avoid the Birthday Attack. Therefore, in this paper we use 128-bits (16-bytes) as block size.

- **Third Step (Substitution Operation):** the substitution operation for each block of the message will be applied. The operation is based on S-Box table, character, index of character in the block (to avoid duplicate character in the same block), and block number (to avoid duplicate character in the different block but the same index of character). Doing such operations is to avoid the cryptanalysis or any attack by producing cipher text to be used to calculate the MAC

The Encryption-then-MAC (EtM) is the best case for message integrity [14]. MAC based on EtM the will be generated from the encrypted message. To generate a cipher text from a given message, we use substitution operation. Therefore, the character, which regards as a column name in the S-Box whiles the row, is the index of the same character. The substitution character sums up with the block number and original character mod 256; to generate the corresponding cipher character.

- **Fourth Step (Mathematical Operation):** use mathematical equation to generate the MAC from the message. The MAC must be depending on the all blocks, therefore the CBC-Mode will used in this step.
  In CBC-Mode set the key as initial vector, in each round the function is summation between each: the byte in previous block, byte in current block, and previous byte in current block. Finally the final block is a MAC (tag) generated. Fig 4 shows the block diagram for proposal methodology.

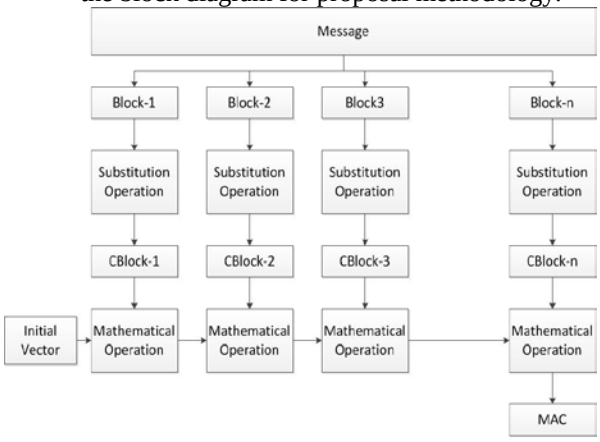

Fig. 4 Block diagram to generate the Message Authentication Code

Pseudo Code for Algorithm (2):

- Initial Message (M), Secret Key (K), 4 Prime Numbers (P), initial-vector (IV).
- S-Box Data = GenerateSBox(k, P), using Algorithm (1).
- Split M to List<Block>.
- For Loop i=0…block count
- For Loop j=0…character count in block
- C-block (i)(j) = SubstitutionOperation(block(i)(j), j, i, S-Box Data).
- End For j
- End For i
- For Loop s=0…C-block count
- IV = Sum (IV, C-block(s))
- End For s
- MAC = IV

The sender algorithm generates MAC based on Eq. 1. Fig 5.a shows the sender procedure.

$$MAC = C\,(K, M) \quad (1)$$

Where:
MAC: Message Authentication Code.
C: MAC function.
K: shared secret key.
M: input message.

Then message and MAC will send to the receiver. The receiver algorithm generates MAC' based on equation (1). Fig 5.b shows the receiver procedure. The message will be verified by comparing/matching the two generated MAC's in receiver side. If any mismatch will be detected, then the message has been altered (e.g. insert character, delete, modify).

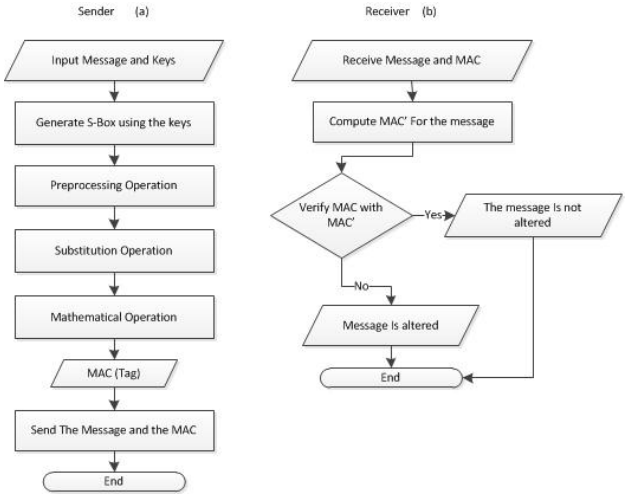

Fig. 5 (a) Flowchart for sender (b) Flowchart for receiver

Example1: The example shows the result for each step of proposal methodology. The input message is "This Message is used in test example to provide Message Authentication Code". The key is "ComputerScience1" while prime numbers are (7933, 8161, 9901, 10009).

1. **Generate the S-Box Step:** generate the S-Box table by using the key and the prime numbers.

2. **Preprocessing Step:** "This Message is used in test example to provide Message Authentication CodeCompu", by added "Compu" to complete the final block, then split message to five blocks, where block size is 16-bytes (128-bits):

Block-1: 84, 104, 105, 115, 32, 77, 101, 115, 115, 97, 103, 101, 32, 105, 115, 32
Block-2: 117, 115, 101, 100, 32, 105, 110, 32, 116, 101, 115, 116, 32, 101, 120, 97
Block-3: 109, 112, 108, 101, 32, 116, 111, 32, 112, 114, 111, 118, 105, 100, 101, 32
Block-4: 77, 101, 115, 115, 97, 103, 101, 32, 65, 117, 116, 104, 101, 110, 116, 105
Block-5: 99, 97, 116, 105, 111, 110, 32, 67, 111, 100, 101, 67, 111, 109, 112, 117

3. **Substitution Step:** The results after substitution operation using s-box are:
Block-1: 210, 150, 81, 221, 156, 196, 130, 10, 242, 228, 126, 25, 250, 110, 14, 72
Block-2: 161, 61, 157, 23, 157, 237, 190, 73, 209, 189, 39, 21, 251, 69, 124, 194
Block-3: 198, 120, 87, 234, 158, 165, 26, 74, 238, 27, 232, 56, 38, 152, 170, 74
Block-4: 223, 139, 250, 224, 123, 241, 133, 75, 233, 235, 184, 155, 29, 9, 179, 51
Block-5: 59, 53, 214, 143, 99, 114, 186, 116, 196, 118, 138, 78, 229, 0, 66, 235

4. **Mathematical Step:** used to generate MAC (tag), the result is:
MAC: 236, 179, 223, 253, 217, 156, 19, 192, 172, 87, 68, 129, 241, 68, 135, 38

5. Requirements and Properties of MMAC

1. Variable length of input message making our algorithm flexible.
2. Fix length of MAC value.
3. Contain the S-Box key.
4. One way algorithm.
5. No two different messages have a same MAC value.
6. MMAC are efficiency and easy to compute.
7. If we change one bit/byte from a message then the MAC value will be changed more than half bytes.
8. Pseudo Randomness value.

6. Result and Discussion

The result shows, without any doubt, that the propose algorithm is very efficient and secure in the following aspects, comparing with well-known literature algorithms:

- **Fast:** the proposed algorithm has faster execution time than standard algorithms. We compare our algorithm with CMAC-AES (Cipher-Based Message Authentication Code-Advanced Encryption Standard), MAC-Triple-DES (Message Authentication Code-Triple-Data Encryption Standard) and CMAC-DES (Cipher-Based Message Authentication Code-Data Encryption Standard) MACs algorithm. And compare with HMAC-SHA-128 (Secure Hash Algorithm) HMAC algorithm. Table 1 and Fig 6 shows the executing time of algorithms.

Table 1: Execute time compared with standard algorithms

no	Algorithm Name	Execution Time	Block Size	Data set Size
1.	HMAC-SHA-128	36.7 ms	128-bit	273 KB
2.	MAC-Triple-DES	52.4 ms	128-bit	273 KB
3.	CMAC-AES	50 ms	128-bit	273 KB
4.	CMAC-DES	35.3 ms	128-bit	273 KB
5.	**MMAC (proposed)**	**26 ms**	**128-bit**	**273 KB**

The significant less time came as a result of using:
1- A single round instead of multiple rounds.
2- A simple substitution encryption instead of using triple DES or AES.
3- Simple mathematical equation (summation of 3 numbers) instead of using complex equations.

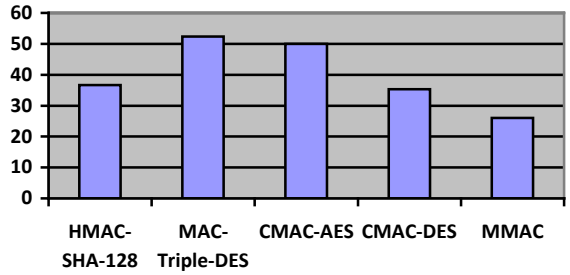

Fig. 6 Execute time compard with standard algorithms

- **Less Complexity:** standard MAC algorithms use very complex AES which lead to very low throughput. In other hand, the proposed algorithm is less complexity by avoiding AES to increase the throughput.

- **Secure:** to demonstrate the security of our algorithm, we test it with some of the most robust attacks in this type of security service and show how to avoid each attack in our algorithm. Below are the high effective attacks and how we provide a better solution:

1- Block re-ordering attack: the attacker exchanges the order of message blocks [15].

Our proposed Solution: authenticate each block based on its index (block number). Therefore, if any attack changes the order block then the MAC will be changed as well, while other methods do not change the MAC.

2- **Truncation attack:** some of MAC algorithms not based on the final block, therefore when the attacker deletes blocks from the end of the message, the MAC value here will not change [15].
Our proposed Solution: while proposed algorithm generates the MAC value based on all blocks (using block number) to avoid any changed.

3- **Mix and Match attack:** the attacker has valid tags (tag1, tag2, tag3) and (tag1', tag2', tag3') on the message (message1, message2, message3) and (message1', message2', message3'). The attacker outputs (tag1, tag2', tag3) on the message (message1, message2', message3) [15].
Our proposed Solution: authentication of each block changes most (more than half) bits of the MAC block.

4- **Brute-force attack:** in this attack, the attacker tries all possible keys to get the correct plain-text that corresponds to the cipher-text. The number of trials depends on the length of the key ($2^{\wedge key\text{-}size}$). This attack is more difficult on MAC than on Hash [3].
Our proposed Solution: in our method we generate a strong and large S-Box Sub-key which depends on secret keys and a robust algorithm. Thus, the number of attack tries will be huge.

5- **Cryptanalysis attack:** here, the attacker tries to deduce a plain-text or key that is taken from the analysis of cipher-text. The analysis is based on the target algorithm and the knowledge of the general characteristics of the plain-text [3].
Our proposed Solution: this attack cannot be happened in our proposed method, because the generation of the cipher-text is based on the S-Box table that has same frequency for each character. In addition, is based on block number, index of character in the block, and the weight of the character.

6- **Birthday attack:** it's also called the birthday paradox attack. The birthday attack against the MAC function with range size R which requires ($R^{1/2}$) trials to obtain the collision block [16].
Our proposed Solution: in our method the key size is 128-bits which means the number of trials to get a collision is very large. Therefore, the sending message is hard to be attacked.

7- **Second pre-image resistance:** it is one of pre-image attacks against cryptographic hash functions. it tries to find another message for the same tag. It also refers to weak collision resistance. The attacker should not find the two messages that have the same tag [4].
Our proposed Solution: in our algorithm we focused on the position of the character in the message, the weight of it, block number, and also used the CBC-Mode that each block depends on the previous block. Therefore, any change in the original message will cause generated a different tag.

8- **Man in the middle attack:** when sender transmits a message plus the tag, the adversary intercepts and modifies the message. Then the attacker will calculate a new tag. While the receiver is not able to detect that the message was modified [3].
Our proposed Solution: in our MAC generator is based on the shared secret key, so the receiver will know when the adversary modifies the message.

7. Conclusion and Future Work

We conclude that the new method (Mathematical Message Authentication Code (MMAC)) is faster and more secure as it is single round, simple mathematical equation, and without using the AES-algorithm. Our algorithm is used to provide integrity of a message (message authentication) and also is used to provide user authentication, which means that the receiver can prove that the received message was sent by authorize sender.

The implementation of this algorithm is efficient and uses the optimize way. However, we can use parallel substitution operation for blocks rather than sequential substitution operation for blocks.

References

[1] Neeta Wadhwa, Syed Zeeshan Hussain, and S. A. M. Rizvi, "A Combined Method for Confidentiality, Integrity, Availability and Authentication (CMCIAA)," in Proceedings of the World Congress on Engineering, vol. 2, 2013.
[2] Erez Petrank and Charles Rackoff, "CBC MAC for real-time data sources," Journal of Cryptology, vol. 13, pp. 315-338, 2000.
[3] William Stallings, Cryptography and network security: principles and practices.: Pearson Education India, 2006.
[4] Keting Jia, Xiaoyun Wang, Zheng Yuan, and Guangwu Xu, "Distinguishing and second-preimage attacks on CBC-like

MACs," in International Conference on Cryptology and Network Security, 2009, pp. 349-361.
[5] Tetsu Iwata and Kaoru Kurosawa, "omac: One-key cbc mac," in International Workshop on Fast Software Encryption, 2003, pp. 129-153.
[6] Mridul Nandi, "Fast and secure CBC-type MAC algorithms," in Fast Software Encryption, 2009, pp. 375-393.
[7] Nicky Mouha et al., "Chaskey: an efficient MAC algorithm for 32-bit microcontrollers," in International Workshop on Selected Areas in Cryptography, 2014, pp. 306-323.
[8] John Black, Shai Halevi, Hugo Krawczyk, Ted Krovetz, and Phillip Rogaway, "UMAC: Fast and secure message authentication," in Annual International Cryptology Conference, 1999, pp. 216-233.
[9] Morris Dworkin, Recommendation for block cipher modes of operation: Galois/Counter Mode (GCM) and GMAC.: US Department of Commerce, National Institute of Standards and Technology, 2007.
[10] Daniel J. Bernstein, "The Poly1305-AES message-authentication code," in International Workshop on Fast Software Encryption, 2005, pp. 32-49.
[11] Kazys Kazlauskas and Jaunius Kazlauskas, "Key-dependent S-box generation in AES block cipher system," Informatica, vol. 20, pp. 23-34, 2009.
[12] Ashwak Mahmood Alabaichi, "A dynamic 3D S-box based on cylindrical coordinate system for blowfish algorithm," Indian Journal of Science and Technology, vol. 8, 2015.
[13] K. Balajee Maram and J. M. Gnanasekar, "Evaluation of Key Dependent S-Box Based Data Security Algorithm using Hamming Distance and Balanced Output," TEM, 2016.
[14] Mihir Bellare and Chanathip Namprempre, "Authenticated encryption: Relations among notions and analysis of the generic composition paradigm," Journal of Cryptology, vol. 21, pp. 469-491, 2008.
[15] Jonathan Katz and Yehuda Lindell, Introduction to modern cryptography.: CRC press, 2014.
[16] Mihir Bellare and Tadayoshi Kohno, "Hash function balance and its impact on birthday attacks," in International Conference on the Theory and Applications of Cryptographic Techniques, 2004, pp. 401-418.

Mohammed Ali received the B.S. in computer science from university of Baghdad in 2013. master student currently. He has patent reward in 2016. He has appreciation letter from the president of Karkh University of science. Holds the first place in the scientific competition sponsored by the ministry of youth and sports held at the University of Baghdad. Also he holds 3 books of thanks and appreciation from the Dean of the Faculty of Science.

Loay K. Abood received the M.S. and Ph. D. degrees in Physics Science from University of Baghdad in 1993 and 1999, respectively. During 2007-2010 work as head of computer science department. Currently he works as Assistant President for Scientific Affairs, Karkh University of science.

Makki Maliki received the M.S. in computer science from University of Jordan in 2003. and received the Ph. D in computer science from university of Buckingham UK in 2015. Currently instructor in the University of Baghdad. Iraq. Interesting Area: Image processing, Pattern recognition, Writer identification, OCR, Medical Image, and Biometrics.