International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 7, No.5 (Sep-2018)

http://dx.doi.org/10.12785/ijcds/070504

Modified Advanced Encryption Standard Algorithm for Reliable Real-Time Communications

Mahmoud A. ELTatar [1] and Ammar M. Abu-Hudrouss[1]

[1] *Electrical Engineering Department, Islamic University of Gaza, Gaza, Palestine*

Received 1 Mar. 2018, Revised 12 Jul. 2018, Accepted 12 Aug. 2018, Published 1 Sep. 2018

Abstract: Ensuring the security of data while being transmitted or stored is a very important issue for companies and end users. Cryptography is used to transfer the data in a form that is not understood by anyone apart from the intended recipients. The advanced standard encryption algorithm (AES) is one of the most secure encryption algorithms. However, due to the speed issue and computational complexity, it is not suitable for real time applications. In this paper, a modified scheme is developed for the encryption/decryption by modifying the MixColumns stage in the AES algorithm in order to increase the speed of the encryption/ decryption while maintaining the complexity level of the design to keep the security level equivalent to AES. The results show that the proposed modification increases the speed of the encryption/decryption in real time applications up to 150% in the encryption process and up to 500% in the decryption process.

Keywords: Advanced encryption standard, AES, Information security, Cryptography, Modified Advanced Encryption Standard, modified MixColumns

1. Introduction

The security of the sensitive information is a prime concern for every business. Modern communication and storage techniques, using computers connected through networks, make the sensitive data vulnerable for many threats. The basic security concepts which are important to sensitive information include: Confidentiality, Data Integrity, Authentication and Non-repudiation [1, 2]. Confidentiality means that the information is accessed only by the authorized people. Integrity indicates no authorized changes are made to the data. Authentication is proving that a user is the person he or she claims to be. Whereas, non-repudiation means that the user cannot later deny that he or she performed the activity. Another important concept is the authorization which is the act of determining whether a particular user has the right to access the data [3].

Security of any communication system is one of the big issues today. Making the encryption algorithms more complex increases the security but it affects the speed of the transmission due to the time consumed in the encryption and decryption processes [4]. Therefore, this solution will not be suitable for some communication links especially for real-time applications that need to be connected online all the time.

The AES algorithm is one of the best algorithms for symmetric cryptography; it uses variable key lengths and can encrypt up to 128 bit data block [5].

In this paper, a modified scheme for AES algorithm is proposed to decrease the encryption and decryption time. The suggested scheme is denoted as MAES. The MixColumns stage, which depends on the polynomials, is replaced by a permutation stage depends on initial vectors (IV). The IV vectors generated for each round from the key expansion process. As the IV vector depends on the key of the encryption, the modified algorithm maintains the complexity of the AES algorithm.

2. Related AES Modifications

Modifying the AES algorithm has been done several times before. Nevertheless, each researcher has limited his concerns to one characteristic to improve. The main characteristics that the researchers concentrated on are the speed of the encryption and decryption processes such in [5] and customization for specific applications like in [6]. In [6], the authors introduced a new scheme to increase the security level for AES. The problem was that the Electronic CodeBook (ECB) mode for AES have a weakness in static data [7]. The scheme depends on

E-mail: mahmod__tatar@hotmail.com, ahdrouss@iugaza.edu.ps

adjusting the shift row stage. This scheme from cryptographic viewpoint is highly secure and gives better encryption results in terms of security against statistical attack.

In [5], the authors introduced a new scheme for AES by replacing the Mixcolumn stage with the permutation step from the DES algorithm. This scheme provides a high-speed encryption and decryption specifically for media like image and suitable for a plaintext transfer.

In [8], the authors upgraded the AES algorithm by increasing the data block size and using a longer encryption key. The used key lengths is 384, 512, 658, or 1024 bits and the data block will be processed as a group of 48, 64, 96, or 128 bytes. This scheme gives the AES algorithm more complexity and more robustness.

In general, the proposed schemes modified the AES algorithm for only one purpose. For example, in [6] the speed of the algorithm kept as it. However, this in not convenient with limited resource applications especially in real-time applications. While in [5], the security level of the algorithm decreases significantly. This is not a good solution for high security requirements, taking into consideration the powerful computational process for the attackers. In [8], the whole structure of the algorithm has been modified by increasing the key length and the data block size. This algorithm is also not suitable for real time applications that use a small number of bits at each sample.

3. THE ADVANCED ENCRYPTION STANDARD (AES)

A. *Introduction*

AES is one of the most used algorithms in cryptography field. It is considered a very complex and secure algorithm. The speed of the encryption and decryption process are less than the other algorithm. However, it is not fit for real-time applications and must be implemented carefully to get the full security performance. AES is a symmetric block cipher. This means that it uses the same key for both encryption and decryption processes. The algorithm Rijndael (chosen to be the winner of AES) allows for a variety of block and key sizes. In fact, the key can be chosen independently from 128, 160, 192, 224, and 256 bits and needs not be the same. However, the AES standard states that the algorithm can only accept a block size of 128 bits and a choice of three keys of 128, 192, or 256 bits, is depending on which version is used [7].

The overall structure of AES can be seen in Figure 1. The plain text data is grouped into blocks of 4 × 4 byte matrix. Each block is then copied into an input state array, which is modified at each round of the algorithm. Finally, the state array is taken as an output block.

The round term indicates a basic process in the encryption and decryption which repeated multiple times depending on the length of the encryption key. Each round consists of 4 stages which are AddRoundKey phase, SubByte phase, ShiftRow phase, MixColumns phase [9]. MixColumns is the stage that replaced by the permutation stage.

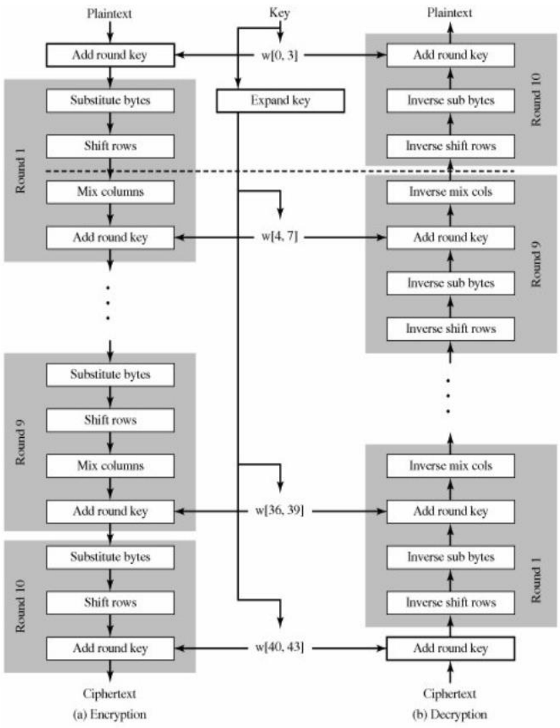


Figure 1. Overall structure of the AES algorithim [4].

B. *MixColumns phase*

In the MixColumns phase, the block matrix is divided into 4 columns; each column is transformed to a fourth order polynomial over Galois Field of size GF(2^8) and multiplied by a fixed polynomial $f(x)$ which is given as,

$$f(x) = (\{03\}x^3 + \{01\}x^2 + \{01\}x + \{02\})\ modulo\ (x^4 + 1) \qquad (1)$$

The multiplication of the input matrix with this equation can be transformed into a matrix multiplication in the following form,

$$\boldsymbol{S'^{(x)}} = \boldsymbol{F(x)} \otimes \boldsymbol{S(x)}$$

$$\begin{bmatrix} S'_{0,c}\\ S'_{1,c}\\ S'_{2,c}\\ S'_{3,c}\end{bmatrix} = \begin{bmatrix} 02 & 03 & 01 & 01\\ 01 & 02 & 03 & 01\\ 01 & 01 & 02 & 03\\ 03 & 01 & 01 & 02\end{bmatrix}\begin{bmatrix} S_{0,c}\\ S_{1,c}\\ S_{2,c}\\ S_{3,c}\end{bmatrix} \qquad for\ 0 \le c \le 3 \quad (2)$$

The complete mathematical implementation of the MixColumns stage can be found in [9].

C. *Key Expansion phase*

The key expansion phase generates a group of sub keys that will be used in the AddRoundKey stage in the encryption process.

4. Modified AES algorithm (MAES)

A. *Introduction*

In the modified algorithm, MixColumns stage is replaced with a permutation stage as shown in Figure 2. The random permutation stage will be done using an *xor* operation between the input state of the permutation stage and a random IV-vector. The vector is generated using random number source at the initializing of the encryption session as shown in Figure 3.

The random number generator has the encryption key as a seed for its process. Such scheme will make the generated IV-vector key dependent so the complexity of the encryption process will increase. However, the time needed for encryption and decryption will decrease because the computational demand for the *xor* operation is much less than the MixColumns stage.

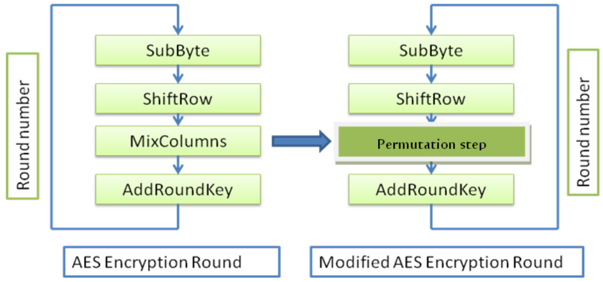
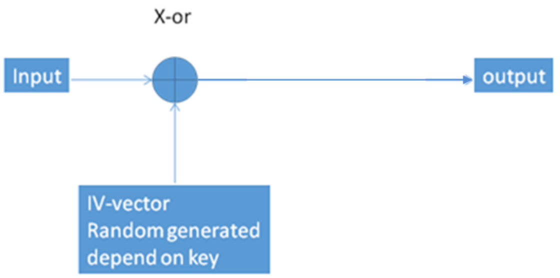


Figure 2. The overall structure of the MAES.



Figuire 3. The permutation stage that replaced the Mixcolums stage.

B. *The proposed algorithm design*

The first goal of the MAES scheme is to increase the speed of the encryption and decryption algorithms. The MixColumns stage is the most computational demanding stage in the AES design and therefore it consumes most of the time needed for encryption and decryption. In the MAES design, the MixColumns stage is replaced with an *xor* operation between the input state and a random vector called IV.

This *xor* operation decreases the calculation demands to only 16 *xor* cycles. This is because the input state has 16 bytes and the *xor* operation consumes one cycle for each byte to be done.

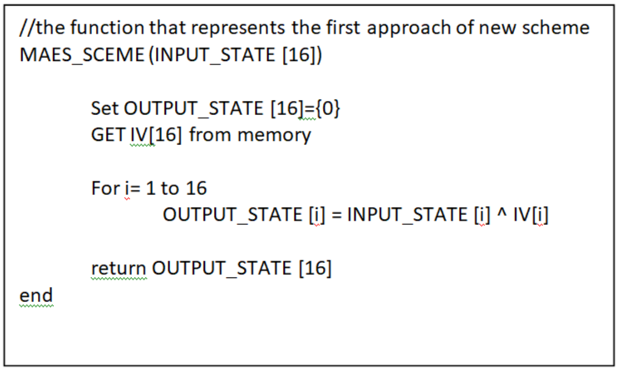
This number of cycles can be achieved using low-cost hardware that depends on an 8-bit processor in the implementation of this scheme. However, this number of cycles can be optimized if the encryption/decryption process is accomplished using hardware with higher specifications. For example, if the dedicated hardware depends on a 32-bit processor, the overall cycle will be reduced to only 4 cycles since 4-*xor* operations can be done in one cycle.

The MAES uses two approaches for the second input of the *xor* operation:

1- One IV vector for the entire rounds in the encryption and decryption processes.

2- One IV vector for each round in the encryption and decryption processes.

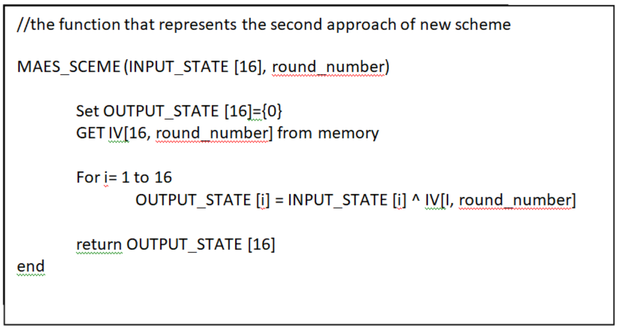
The first approach will increase the speed of the initialization at the start of the encryption session. While the second will increase the security complexity for the modified scheme. The pseudocode of the first approach of the MAES is shown in Figure 3.



```
//the function that represents the first approach of new scheme
MAES_SCEME (INPUT_STATE [16])

        Set OUTPUT_STATE [16]={0}
        GET IV[16] from memory

        For i= 1 to 16
                OUTPUT_STATE [i] = INPUT_STATE [i] ^ IV[i]

        return OUTPUT_STATE [16]
end
```

Figure 3. Pseudocode for the first approach of the MAES.

While, the pseudocode of the second approach of the MAES is shown in Figure 4.



```
//the function that represents the second approach of new scheme

MAES_SCEME (INPUT_STATE [16], round_number)

        Set OUTPUT_STATE [16]={0}
        GET IV[16, round_number] from memory

        For i= 1 to 16
                OUTPUT_STATE [i] = INPUT_STATE [i] ^ IV[I, round_number]

        return OUTPUT_STATE [16]
end
```

Figure 4. Pseudocode for the second approach of the MAES.

As shown in Figure 3 and 4, the input for the first approach is the input state. While the input of the second approach is the input state and the round number but in the two cases the estimate time is similar.

To increase the security level, the second design will be chosen for the following reasons:

1- It is more complex with similar time consumption. (As it needs 16 IV vectors in the second approach while in the first one it needs only one IV vector).

2- The extra time needed for the generation is small as the operation is done only once at the initialization step.

C. The proposed random number generator

The second goal for MAES is to maintain the complexity of the algorithm as high as possible. Therefore, the MAES uses two different algorithms for random number generation to generate the IVs vectors that are needed in the encryption process.

The first algorithm is to extend the key expansion algorithm with modified rounds to generate the IVs vectors. This algorithm uses an external random number generator to generate the IV vectors.

The second algorithm is to use the key expansion algorithm as a random number generator. As a result, this decreases the memory needed for the MAES in comparison with external random number generators algorithms that need a high memory allocation.

However, this approach in generating the random number IV vectors is not secure. This is because the key expansion algorithm is not considered as a complex random number generator due to the simplicity of its design. Therefore, in order to increase the complexity of the MAES, an external random number generator can be used.

An external random number generator will allow generating a complex random numbers (pseudo RNG) [10] that needed to generate the IV vectors. This will increase the security of the MAES.

The new design uses the "*sha3*" which is one of the HASH functions [10] as a random number generator. This is illustrated in Figure 5. The pseudocode of the algorithm that uses the *sha3* function to generate the one IV vector is shown in Figure 6. While the pseudocode of the *sha3* function to generate the 16 IVs vectors is shown in Figure 7.

The HASH functions have a high complexity and high speed. This encourages using them in the MAES as IV-vector generator [11].

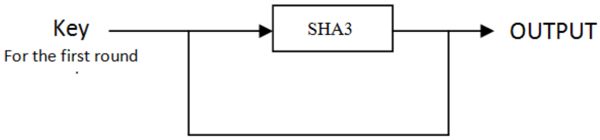
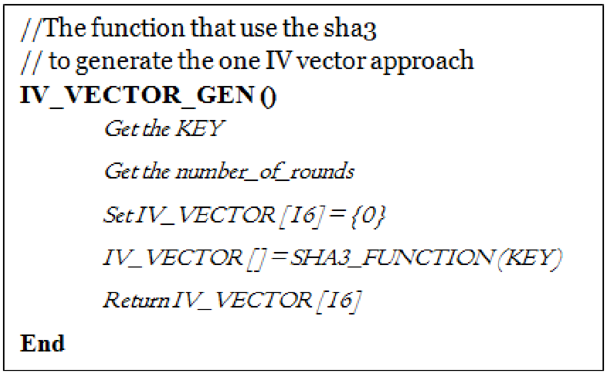
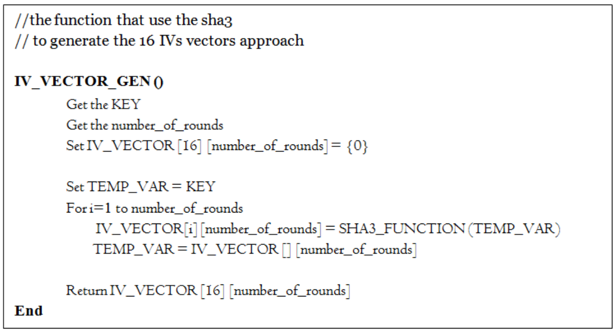


Figure 5. *SHA3* as an external random number generator.



```
//The function that use the sha3
// to generate the one IV vector approach
IV_VECTOR_GEN ()
        Get the KEY
        Get the number_of_rounds
        Set IV_VECTOR [16] = {0}
        IV_VECTOR [] = SHA3_FUNCTION (KEY)
        Return IV_VECTOR [16]
End
```

Figure 6. Pseudocode for using *SHA3* to generate one IV vector algorithm.



```
//the function that use the sha3
// to generate the 16 IVs vectors approach

IV_VECTOR_GEN ()
        Get the KEY
        Get the number_of_rounds
        Set IV_VECTOR [16] [number_of_rounds] = {0}

        Set TEMP_VAR = KEY
        For i=1 to number_of_rounds
            IV_VECTOR[i] [number_of_rounds] = SHA3_FUNCTION (TEMP_VAR)
            TEMP_VAR = IV_VECTOR [] [number_of_rounds]

        Return IV_VECTOR [16] [number_of_rounds]
End
```

Figure 7. Pseudocode for use *SHA3* to generate the 16 IVs vectors algorithm.

5. EXPERIMENTAL RESULTS

The modified scheme was implemented and the result proves that the efficiency of the MAES. The simulation is done on a desktop computer device with core i7 with 2.4 GHz clock.

At the initial stage of encryption, the modified key expansion phase result in table 1. It is clear that the original algorithm is faster than the modified because the introduced *xor* operation in the new scheme.

TABLE I. KEY EXPANSION TIME CONSUMPTION

State	Time (*sha3*)	Time (key expansion)
AES (original algorithm)	8.3 us	8.3 us
Approach 1: One IV vector generation for all rounds	17 us	26.5 us
Approach 2: One IV vector generation for each round	70 us	29.1 us

The result in Table I shows that using an external random number generator will increase the time needed to generate the IV vector in the first approach to the twice of the time needed in the original AES Key expansion algorithm. However, this time is less than the time needed in the modified AES key expansion algorithm, which includes the original AES key expansion process. On the other hand using the external approach will increase the time needed to seven times greater than the AES algorithm and double the time needed for the modified key expansion algorithm. Needless to say that this process is done at the initialization and has a little effect on the total consumed time. Table II shows the time consumption for the MixColumns stage compared with the permutation stage in the MAES.

TABLE II. ENCRYPTION AND DECRYPTION TIME FOR THE MIXCOLUMNS AND PERMUTATION

Algorithm	Encryption	Decryption
Original AES	0.06 uS	2.36 uS
Approach 1: One IV vector generation for all round	0.03 uS	0.04 uS
Approach 2: One IV vectors generation for each round	0.03 uS	0.03 uS

In Table II, the difference between the two algorithms is clear. When using an optimized implementation for the MixColumns stage the time is reduced to the half in the encryption process. However, the decryption process has more complex operations in the MixColumns stage. Therefore, the time needed for the permutation stage is lower by 50 times compared to the MixColumns stage.

The overall time consumption for the encryption and decryption process is shown in table III.

TABLE III. ENCRYPTION AND DECRYPTION TIME FOR ONE BLOCK ENCRYPTION

Algorithm	Encryption	Decryption
Original AES	74 uS	253 uS
Original AES (optimized implementation)	34 uS	50 uS
Approach 1: One IV vector generation for all round	31 uS	31 uS
Approach 2: One IV vectors generation for each round	30 uS	30 uS

In this table, the time of the encryption decreases to the half when using the AES implementation while the decryption time decrease to on- fifth.

These results are a little worse when using optimized implementation for the AES algorithm. The optimized implementation depends on assumptions found in the hardware used to decrease the time needed for the Mixcolumns stage. Therefore, the encryption time for the MAES decreases slightly and decryption time decreases to the half.

The overall results show that even when using an optimized implementation, which cost more memory, the MAES is still the fastest.

6. SECURITY ANALYSIS

A. *introduction*

To ensure the security of the encrypted data and the encryption algorithm the MAES must pass several security tests. Here, the following three tests will be examined:

1- The key space analysis.
2- The histogram statistical analysis.
3- The information entropy analysis.

B. *Key space analysis*

The key space analysis depends on the encryption key length [12]. It also depends on how much the attacker need to decrypt the data using a brute force attack (which depends on trying all the possible keys to decrypt the data) [13]. For example, if the key length is 3 bits then the attacker need to try only 8 keys (2^8) to break the encryption. In the new algorithm, the key length is still as the original algorithm so the key space to perform the brute force attack is (2^{128}) or (2^{192}) or (2^{256}) depending on the key size. Such combinations make the brute force attack invisible so this is a reliable implementation for a practical use.

C. *Histogram Statistical analysis*

Histogram is a statistical test that describes how the bytes distributed [14, 15]. In a file including a plaintext (not encrypted data), there is some duplicated bytes and this can leak some sort of information about the data. From this point, any encrypted data must have equally distributed bytes so no information is leaked from the histogram of the encrypted data. The histogram for the new scheme is shown in figure 8.

Figure 8 shows the difference between the histogram of the original image before the encryption and after the encryption process. The original image has some peaks that gives some details about the image nature.

These include repeated colors and repeated patterns in the data. The histogram for the MAES shows equal distribution for all bytes of the data due to encryption algorithm random number factor which leads that the output data to be as random as possible. Hence, no data can be leaked in this scheme.

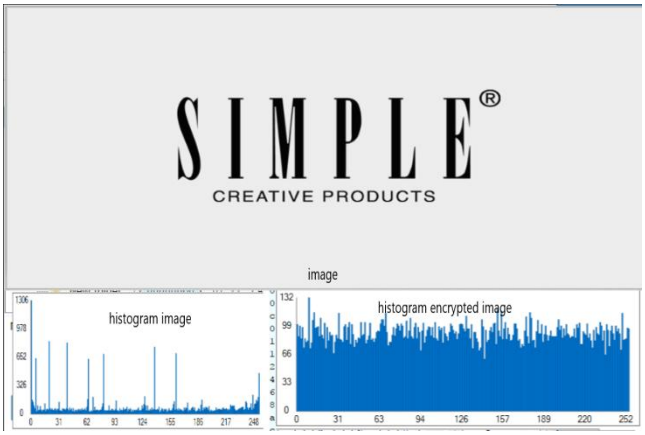


Figure 8. The histogram for non-encrypted and encrypted data.

D. *Information entropy analysis*

The entropy is a measure that indicates the randomness level of the data [16]. For security purpose, the output of the encryption algorithm must appear as random as possible [17]. For the 8-bit algorithm base (one byte), the entropy must be 8 to indicate a complete randomness for the data. For the modified algorithm as well as the original AES, the entropy is less than 8 by a small margin. The entropy for the new scheme compared with the original AES algorithm shown in table 4 proves that the new scheme still save the entropy of the file while the 16-IV scheme is better than the original.

TABLE IV. ENTROPY FOR THE NEW SCHEME AND THE ORIGINAL

Algorithm	**Entropy**
Original AES (Original implementation)	7.9941
Approach 1: One IV vector generation for all round	7.9936
Approach 2: One IV vectors generation for each round	7.9946

7. CONCLUSION

Companies use Cryptography and encryption to protect the sensitive data from unauthorized access or modification.

One of the wildly used algorithms is advance encryption slandered (AES), this algorithm suffers from consuming unnecessary time to achieve the complexity requirements. This hinder using it in real time applications.

Several modifications have been done on the algorithm to reduce the consuming time or to increase the complexity of the algorithm. However, all the modifications concentrate on one purpose which is to decrease the consumption time or to increase the complexity of the algorithm.

In this paper, a new modification is applied on the AES algorithm to adapt the AES algorithm for real-time applications. The new algorithm which is called MAES can increase the speed of the encryption and decryption processes while maintaining the complexity of the encryption as high as possible.

The modification can increase the speed of the algorithm by replacing the Mixcolumns stage with *xor* operation; while the complexity requirements are achieved using random number generator depends on the key to generate the IV vector. The results verify that the modified scheme is faster than the standard AES algorithm while preserving the complexity level.

This MAES will be a good step in the next generation of applications that considers both speed and complexity as critical issues such as blockchain algorithms.

REFERENCES

[1] C. Treacy and F. McCaffery, "Data Security Overview for Medical Mobile Apps Assuring the Confidentiality, Integrity and Availability of Data in Transmission," 2017.

[2] R. Sharma and R. Singh, "An Improved Scoring System for Software Vulnerability Prioritization," in *Quality, IT and Business Operations*: Springer, 2018, pp. 33-43.

[3] P.linda, *Introduction to Information Security, Carnegie Mellon University*. 2008.

[4] A. Roy, A. Misra, and S. Banerjee, "Chaos-based image encryption using vertical-cavity surface-emitting lasers," *arXiv preprint arXiv:1705.00975*, 2017.

[5] H. Sumira, R. Faisal,M. Riaz, A. Gulraiz, A. Anil, D. Abdul ,Modified Advanced Encryption Standard For Text And Images, Computer Science Journal Volume 1, Issue 3, December 2011

[6] S. Abdulkarim , H. Bahaa Eldin, H. Abd El Fatah ,An Efficient Modified Advanced Encryption Standard (MAES) Adapted for Image Cryptosystems, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.2, February 2010

[7] J. Daemen and V. Rijmen, *The design of Rijndael: AES-the advanced encryption standard*. Springer Science & Business Media, 2013.

[8] l. SCRIPCARIU , M FRUNZĂ , Modified Advanced Encryption Standard, 11th International Conference on DEVELOPMENT AND APPLICATION SYSTEMS, Suceava, Romania, May 17-19, 2012

[9] S. William, *Cryptography and Network Security Principles and Practices*, Fourth Edition ed. 2005.

[10] M. J. Dworkin, "SHA-3 standard: Permutation-based hash and extendable-output functions," 2015.

[11] J. E. Gentle, *Random number generation and Monte Carlo methods*. Springer Science & Business Media, 2013.

[12] J. A. E. Fouda, J. Y. Effa, S. L. Sabat, and M. Ali, "A fast chaotic block cipher for image encryption," *Communications in Nonlinear Science and Numerical Simulation,* vol. 19, no. 3, pp. 578-588, 2014.

[13] T. Henk, *fundamentals of cryptology A Professional Reference and Interactive Tutorial*. KLUWER ACADEMIC PUBLISHERS, 2000.

[14] C. E. Lee, L. Zheng, Y. Zhang, and V. L. Thing, "Low-Dimensional Bigram Analysis for Mobile Data Fragment Classification," in *SG-CRC*, 2017, pp. 129-142.

[15] A. A. Shtewi, B. E. M. Hasan, and A. Hegazy, "An efficient modified advanced encryption standard (MAES) adapted for image cryptosystems," *IJCSNS International Journal of Computer Science and Network Security,* vol. 10, no. 2, pp. 226-232, 2010.

[16] M. Edman, A. Kiayias, Q. Tang, and B. Yener, "On the security of key extraction from measuring physical quantities," *IEEE Transactions on Information Forensics and Security,* vol. 11, no. 8, pp. 1796-1806, 2016.

[17] P. J. P. Christof, *Understanding Cryptography A Textbook for Students and Practitioner*. springer, 2010.

Mahmoud A. El-Tatar was born in Gaza city, Palestine, in 1989. He received the B.Sc. degree in Electrical Engineering from Islamic University Gaza, Palestine, in 2012. He received the M.Sc. degree in Telecommunication Engineering from Islamic University Gaza, Palestine, in 2018. He is currently a researcher at Islamic University of Gaza, Palestine. His current research interests are Encryption Algorithms, Communication System, and Microcontroller Systems.

Ammar M. Abu-Hudrouss was born in Khan-Younis, Palestine, in 1977. He received the B.Sc. degree from Islamic University Gaza, Palestine, in 1995. He received the M.Sc. degree in Telecommunication Engineering and the Ph.D. degree in Communication Engineering from Birmingham University, Birmingham, U.K., in 2003 and 2007, respectively. He was a visiting researcher at University of York from 9/2012 to 9/2013 as a holder of Distinguished Scholar Award from the Arab Fund for Social and Economic Development. He is currently an Associate Professor at Islamic University of Gaza, Palestine. He is also a member of CTRC, P-ICTRA. His current research interests are Index Modulation and Coding.