Security Scheme Enhancement for Voice over Wireless Networks

Firas Hazzaa[1,*], Antesar M. Shabut[2], Nada Hussein M. Al i[3], Marcian Cirstea[1]

[1] Faculty of Science & Engineering, Anglia Ruskin University, Chelmsford, UK
[2] Department of Computer Scince, Leeds Tirnity University, Leeds, UK
[3] Collage of Science, University of Baghdad, Iraq

[*]Firas.hazzaa@pgr.anglia.ac.uk

Abstract

This paper proposes a lightweight and low energy encryption algorithm to secure voice traffic over wireless networks. The proposed algorithm meets the requirements of voice traffic and is suitable for wireless devices. It is capable to reduce the execution time and power consumption of the encryption process compared with the state of the art standard algorithm and at the same time maintains the desired security (confidentiality) level. The proposed algorithm employs similar methods with those used in the Advanced Encryption Standard algorithm (AES), with enhancements considering the limitations of wireless devices. A range of simulation scenarios are developed to test the validity of the proposed algorithm in reducing delay and energy required for the encryption process, besides, not compromising the security level. The results show significant improvements in the network and security performance metrics. A comparison between the proposed algorithm, the AES and other lightweight algorithms proposed by other research works is conducted and the results are promising. Using the proposed algorithm, a significant amount of time and energy consumption reduction is achieved to reach approximately 35% enhancement over the standard AES algorithm accompanied with good-level of complexity in the encryption process, making it more suitable for the wireless environment.

Keywords: wireless devices, secure communication, voice encryption, AES encryption, wireless network security

1. Introduction

A wireless network is a set of wireless nodes that connect with wireless links and are not linked by cables of any kind. The use of wireless networks enables enterprises to avoid the costly process of introducing cables into buildings or as a connection between different equipment and locations [1] and [13]. Wireless networks present several advantages such as accessibility, mobility, productivity, deployment, expandability and cost [6], [25] and [39]. Moreover, voice over wireless networks and more generally multimedia applications are very relevant and play an important role in enabling people to communicate with convenience. With the advantages come the challenges which result in making wireless platforms a popular topic of research due to open challenges such as mobility of devices and constrained resources. However, security of the connections between devices and networks is the major challenge in such networks [7]. Nowadays, it is illegal to breach any type of security and privacy of users, and it is hugely restricted by the law, besides the lost and cost, it caused for organizations. According to Steve Morgan in [14], the cost of cyber security reached $84 billion in 2015 and it is expected to reach $170 billion in 2020. The important challenge for supporting audio

applications in the wireless networks is finding a security algorithm that meets the requirements of those applications like delay, and at the same time is suitable for wireless devices which have limited battery energy. That is any encryption enforcement should be lightweight and secure. One of the principles of security is confidentiality, which can be achieved by employing the concept of encryption. Encryption algorithms have been used in many applications and protocols to encrypt data transferred into the network as well as encrypt the IP addresses. Besides, the routing information and request-response packets between wireless nodes could also be encrypted. These features have made the encryption very important in security.

Conventional multimedia traffic management algorithms have been developed mainly to satisfy delay and distortion constraints while usually neglecting security requirements which make them much more susceptible to attacks like eavesdropping and data tampering [37]. On the other hand, internal constraints in wireless devices, like limited computational and power capabilities, make the implementation of current security and cryptography algorithms very difficult. Encryption algorithms can provide a good level of security over wireless devices but they consume a significant amount of computing resources such as CPU time and battery power in mobile devices [2]. Furthermore, voice traffic over wireless channels is critical, because it is more sensitive to delay. For example, encrypting an audio file by mobile phone consumes more battery energy and causes the battery to die rapidly [23]. Therefore, developing a lightweight encryption algorithm that enhances the execution time and energy consumption and maintains the desired level of confidentiality in voice traffic is still an open and challenging task. Again, there is always a trade-off between security and performance, so this research aims to provide a suitable balance between these two aspects. Therefore, this work is based on the following research question: *How can the security cost (delay, energy) be reduced without affecting the security level of cryptography implementation for voice over wireless devices?*

In this paper, the development of a new encryption algorithm that enhances the encryption execution time and energy consumption and maintains the desired level of security is presented. The work is based on the Advanced Encryption Standard (AES) algorithm with an improved way of conducting the encryption process. The proposed improvements result in making this algorithm lightweight and suitable for voice traffic over wireless networks (VoWNet), as well as helping to obtain a trade-off solution between security and performance metrics. By modifying the way the AES algorithm's functions work, the main objectives are to propose a new multi S-box function for increasing the security of the encryption and to modify the mix column function to reduce the encryption cost. Eventually, nine rounds iteration is proposed to replace the ten rounds implemented in the original algorithm.

The main contributions of this paper can be summarized as follow: 1) address the security and performance balance issue by proposing a new cryptography scheme for securing real-time voice traffic over wireless networks; 2) the proposed scheme is suitable for wireless devices and meets their requirements in terms of execution time and energy consumption, and at the same time balances the trade-off between security and privacy of the information; 3) conduct the security analysis using various scenarios for both standard AES and the proposed algorithm to measure their strengths and resistance to different kind of attacks.

The paper is organised as follows. Section 2 illustrates the related work in the literature. The proposed framework which includes details of the encryption algorithm is demonstrated in section 3. Section 4 provides the experiments and results with a comprehensive security analysis. Finally, conclusions and future work are presented in section 5.

2. Literature Review

Several research works have been conducted to address security issues related to real time traffic over wireless networks. Some of them focus on developing algorithms that suit dealing with real time limitations such as delays while some others focus on the complexity of the security algorithm and neglect the performance (encryption delay, Energy) requirements. This work expands our previous work [20] which developed a new lightweight encryption scheme. The main contribution of this paper is to conduct further security analyses which are missing in previous works and to test brute force attack on the newly proposed algorithm. In addition, it provides further explanations (based on further literature analysis) considering the wireless environment and its relation to the cryptosystem issues. The research conducted by [7] uses the notion of multipath which is useful for a reliable mobile ad hoc network (MANET). The authors propose the basis for sending secure real time streaming by using a multipath mobile ad hoc network. This technique can offer security for a wireless ad hoc routing and multimedia transfer. They consider a digital signature and an encryption technique and their results show good performance. They also suggest in their future work to use multicast technique which is important in many video applications. However, the limitation of their work is that in each path they are still using current cryptography algorithms which is not computationally practicable to be executed in the mobile environment.

Authors in [3] propose a new encryption/decryption algorithm based on AES scheme. They suggest some modification on two AES functions to increase the security of the encryption technique, keep the execution time at the same level, and maintain the QoS of the real-time traffic. Despite the high-security be achieved in their work, it did not reduce the run cost of the encryption process and consequently, it is still unsuitable to be used to secure wireless networks and devices because it kept the execution time in the same level. In [16], a Symmetric Dual key Dynamic block algorithm (SDD) for digital video in a partial encryption technology is proposed. This algorithm meets the requirements of the real time traffic with high level of complexity at a considerable speed. However, there is still a high energy consumption in their proposed work which not suit the wireless limitations, also, there is a lack of vital security analysis to prove its strength. Moreover, authors in [15] suggest a 5-rounds AES encryption algorithm for multimedia and real time applications in wireless sensor network. The results show a reduction of the execution time. However, there is a risk of hacking because five rounds of encryption is very vulnerable to cryptanalysis.

Another study is published by Das et al. in [26] to propose a new algorithm to generate random S-Box and its inverse S-Box based on using different irreducible polynomial in the finite field GF(2^8) contrarily to only fixed polynomial is used in the original AES standard. The irreducible polynomial in the AES standard is $m(x) = x^8 + x^4 + x^3 + x + 1$, this polynomial is used to find the multiplicative inverse which is well known to any attacker. To overcome this problem in their proposed algorithm, different irreducible polynomial is used every time in the finite field of GF(2^8) and send this to the receiver joint with the secret key to raise the security of the cipher operations. They try to address this security concern, and their results kept the execution time and energy the same. However, this technique may not suit the new wireless environment because it does not reduce the run cost of the encryption process.

Bansod et al. in [33] design a new lightweight compact encryption system based on bit permutation instruction group operation (GRP). They propose a new hybrid system that offers more compact results in terms of memory space and gate equivalents in embedded security. The authors claim that the standard algorithms such as DES & AES have a vast memory requirement and

would not be possible to be implemented for an embedded system scheme. However, the lack of crucial security analysis is missed in their research.

However, all of the aforementioned literature is not suitable for securing real time applications over wireless networks because of many sensitive requirements including delays, throughput and power consumption in wireless nodes. The fact is that there is always a trade-off between the performance (encryption delay, Energy) and the security (confidentiality) level. In the encryption algorithms and network security, high level of security requires more processing time and consumes more energy and vice versa; so developing encryption algorithms to meet the network requirements is still in its infant stages and requires a huge number of research studies. Therefore, this paper takes an effort to address this challenge efficiently by proposing a new algorithm to make it suitable for wireless network environments.

3. Proposed Framework

A lot of evidence can be clearly seen across the globe of the threat of cyberattacks on people privacy and security. Besides, the literature review has provided considerable evidence that clearly showed the security requirements for the next generation of the wireless networks and their limitations. In addition to the security protection, the execution time and energy consumption should be maintained as well, especially for real-time applications. Therefore, balancing the security and, performance such as encryption time and energy consumption, requirements is a big challenge which should be urgently addressed in the research. In this work, a quantitative research method has been adopted in which it involves running two cryptography experiments; the AES algorithm [6] and new algorithm proposed in this paper for audio files with different sizes. The proposed algorithm has been developed and tested using network performance metrics includes delay and energy consumption. Usually in the network environment the encryption occurs in the nodes, so this paper focuses on the encryption process in the nodes and the effect it has in terms of delay and power consumption. To achieve these objectives, the AES algorithm is selected, analysed, and improved to suit the real time traffic and wireless networks environments. Despite AES has proved its strength and suitability for many security applications, it cannot satisfy the requirements for real time applications and wireless nodes limitations [15], [4] and [3].

3.1 Advanced Encryption Standard (AES)

The AES is the most used encryption algorithm [6] and it is recommended by NIST to replace DES technique. It is composed of four transformation functions; AddRoundKey, Sub-Byte, MixColumns and Shift-Rows. It is an iterative algorithm, contains different rounds which are dependent on a key size [12]. There are 10, 12 and 14 rounds used for key length of 128, 192, and 256 bits; default 256. In AES every plaintext encrypted block consists of 128 bits, known as state matrix and is illustrated through a 4x4 bytes square matrix [22] and [42]. It has been tested on a different range of platforms and used for many security applications [38]. The illustration of the AES algorithm that has ten rounds/iterations is shown in Fig. 1 [16] and the description of each function is illustrated as follows.

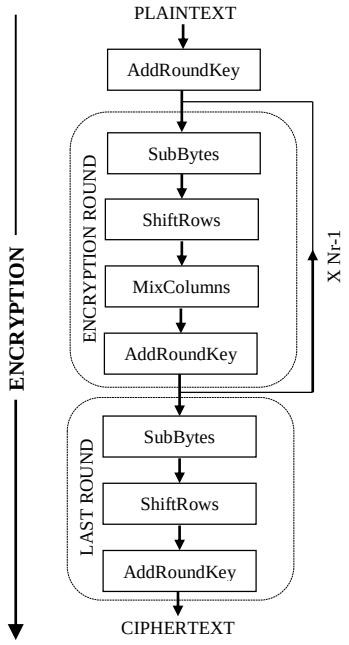


Fig. 1 Standard AES Algorithm

- AddRoundKey: This is a simple function and the most basic form for users. It uses a simplistic bitwise XOR operation. Every 128-bit round key XOR with 128-bit state matrix.
- Sub-Byte: In the Sub-Bytes step, each byte in the state matrix is replaced with a Sub-Byte using an 8-bit data from the Rijndael S-Box. In the inverse Sub-Bytes step, each byte in the cipher matrix is replaced with corresponding inverse Sub-Byte. Sub-Byte operation provides the non-linearity in the cipher. The Sub-Byte process can be visualized in Fig. 2 [7].

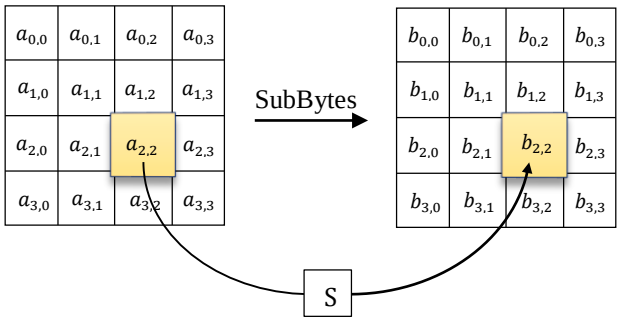


Fig. 2 Sub-Byte Transformation

- Shift-Rows: The transformation during the Shift-Rows stage is a permutation function which enables cycle shifts for each row within the state matrix. This means that the top row of the state matrix remains the same, whereas the other rows are moved in relative cyclical patterns [47].
- MixColumn transformation is one of the more complex functions which makes up the AES. This transformation is applied to the state matrix in a column-by-column manner.

3.2 Voice and Multimedia Traffic

With the growth of the internet, the security of information is gaining more and more interest. The encryption techniques can efficiently safeguard people's information transmitted over public channels. However, the classical encryption systems have restrictions in encrypting such as low efficiency, bulky data, and the high correlation between samples and so on [52]. The fact that the quality of service (QoS), such as encryption time and energy consumption, must be met to provide high-quality media and low latency even when securing media transfer. To provide a secure area as much as possible, an appropriate security algorithm should be chosen in order to send multimedia information in real time due to the unique characteristics of real-time multimedia data such as large data size, high bandwidth and real-time requirements [53].

This section aims to study and investigate the wireless network to identify their traffic behaviour and limitation. The encryption process in networks environment takes place in the nodes (i.e. wireless devices), such as sender or router and does not happen in transmission links between the nodes. Therefore, the delay time for transmitting the data between the sender and receiver is the same for both encrypted and unencrypted data because their size is still the same [6], [9] and [54]. The encryption/decryption time is usually added to a total end to end delay between the nodes in the networks. In this work, the focus is on the execution time at the node where the encryption process is executed (offline scenario). Consequently, the focus should be on the execution time and power for the encryption process needed in the node. Fig. 3 demonstrates the nodes in the wireless network where the encryption executed.

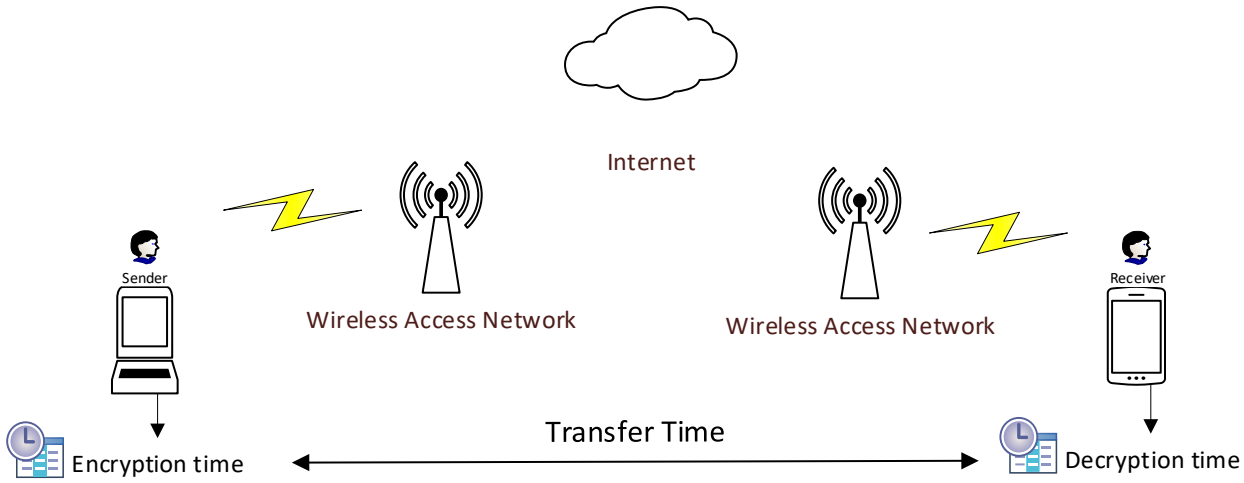


Fig. 3 Encryption and Decryption Time over Wireless Network

Studying the characteristics of real time traffic and their requirements in wireless networks is very important before any security method is proposed. For instance, performance measures such as delay, throughput, and network load should be carefully checked for different traffic types such as voice, video, and text. This enables decision making on the selection of security methods which could be used, without affecting the QoS for these networks. According to Albonda, a distinctive difference in voice packet delay can be noticed when more than 35 nodes participate in mobile ad hoc networks or MANETs. Fig. 4, shows how each mobility model affects the delay in different ways and in relation to the number of nodes in the MANET cluster [11].

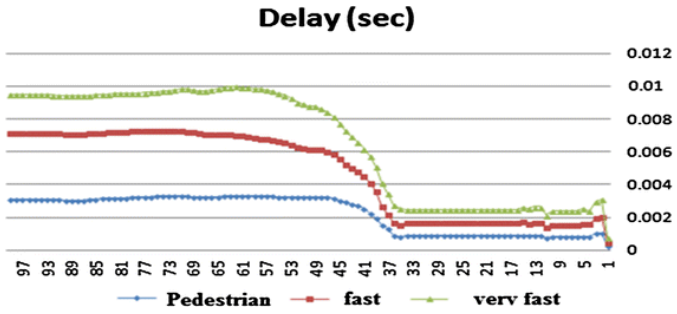


Fig. 4 Packet Delay (in sec) Versus the Nodes Volume [11]

Our previous investigation in [29] has found that the voice applications are more sensitive to delay especially during increasing of participating nodes. Consequently, it should have further consideration when selecting the security applications. Fig. 5 show the average end to end delay for the real time traffic for voice and video conference.

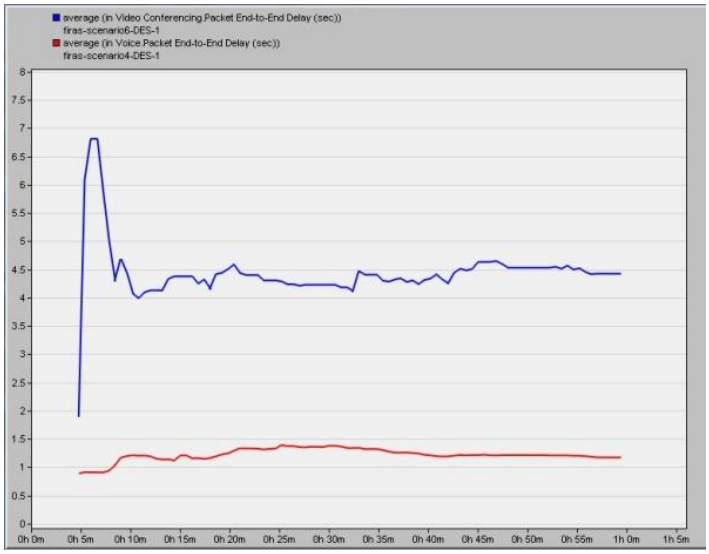


Fig.5. Packet Delay (in sec) Versus the Nodes Volume

In the proposed scheme, it is essential to consider these sensitive requirements for real time traffic and it should make the trade-off between the security level and QoS requirements. And it should offer to the network designer a range of cryptography options, depending on their network density and traffic requirement. For instance, in a small network it may use the encryption algorithm with 20% energy saver, while in a large wireless network it urgently needs more than 50% energy saver (i.e WSN). So the proposed algorithm should consider these facts.

3.3 The Proposed Algorithm

The proposed algorithm uses similar functions with the ones used in AES but with enhancements in order to keep the desired security level of the algorithm and meet the QoS requirements. During the modification of these functions, the focus is on improving the delay and power consumption metrics and maintaining the same or improved level of security. Our solution proposes the modification of the Sub-Byte and MixColumn functions. Sub-Byte function enhancement aims to increase the

security and the complexity of the AES algorithm and to keep the execution time approximately the same, by modifying the Sub-Byte transformation in AES. The MixColumn enhancement, in addition to suggesting nine-round iteration for algorithm processes, aims to decrease the power consumption and execution time and keep the security and complexity at the same level as the original AES. Fig. 6 illustrates the research design and shows the algorithm steps while the following subsections describe the modified functions.

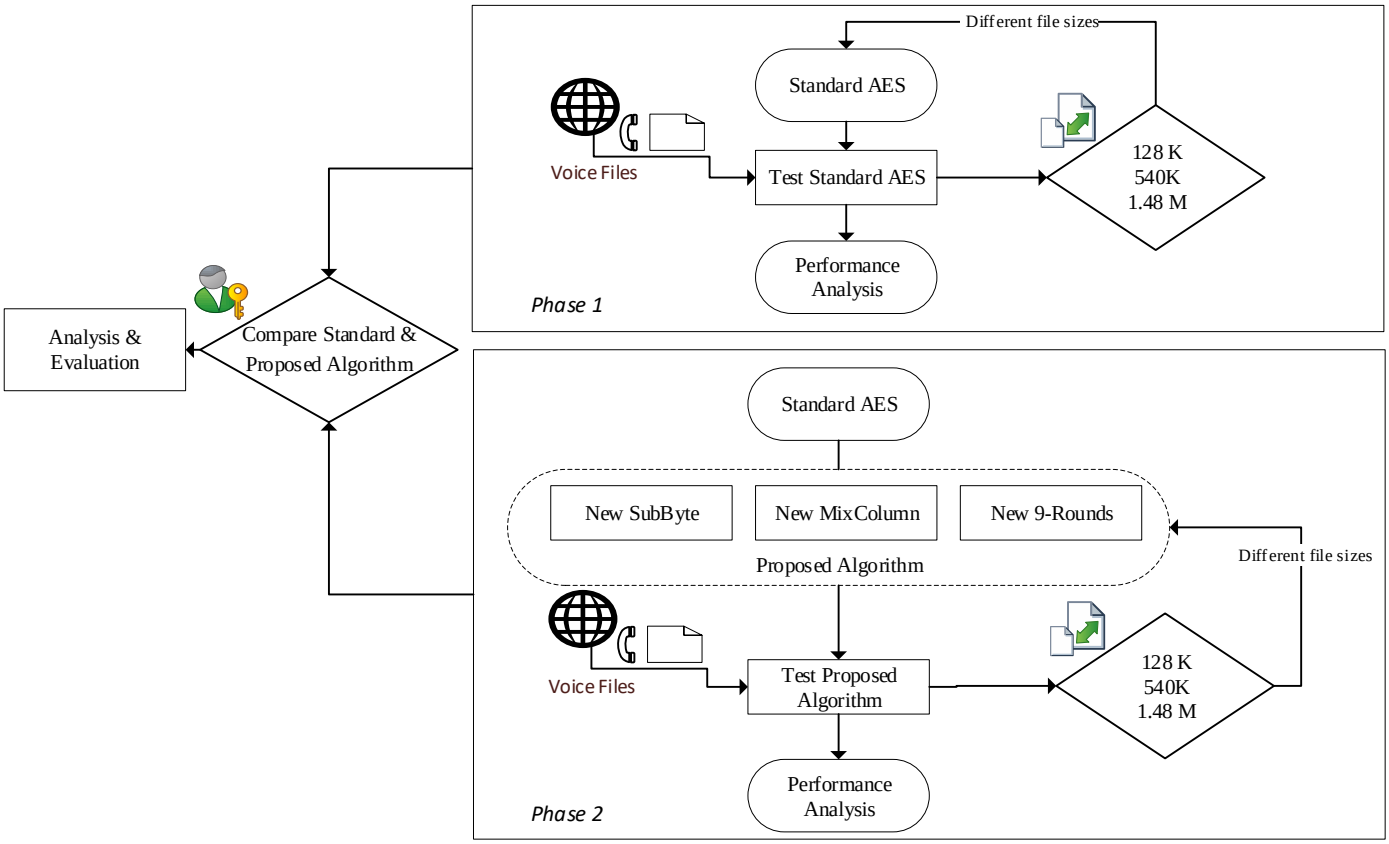


Fig. 6 Research Design and the Proposed Algorithm Steps

3.3.1 Implement Sub-Byte Transformation

This function uses multi s-box transformation instead of one single s-box to increase the security level of this function. According to [3] who propose to use a multi S-Box in the transformation process for the Sub-Byte function using dual keys instead of a fixed structure for the S-Box used in the AES. In this section we are going to implement this technique to prepare for further development in the encryption method. The aim is to make the algorithm more secure, so it is very important to increase the complexity level in the Sub-Byte layout. Using a single S-box is always vulnerable to cryptanalysis method to conduct various attacks [5]. The Sub-Byte function solves the problem of the fixed structure which will lead to the generation of more secure block ciphers. Each byte in the state matrix will be encrypted using different S-Box tables created by the first key, and this in turn increases the security of the AES block cipher system, because multi S-Box function provides non-linearity and would be more difficult for the intruder to analyse the output using a linear equation [5],[3]. A typical single S-Box alone does not have much

cryptographic strength. The key benefit of the proposed function is that several of S-Boxes could be generated. The second key represents a random distribution of the S-Boxes created by the first key. This key will be in the form of a set of sequence S-Boxes tables arranged randomly, chosen by the two parties (i.e. the sender and recipient). This operation leads to an increase of the degree of complexity within the same delay time during the encryption and decryption processes in the proposed Sub-Byte function. Algorithms 1 and 2 demonstrate the aforementioned processes. Fig. 7 illustrates how the enhanced Sub-Byte transformation uses multi S-Boxes.

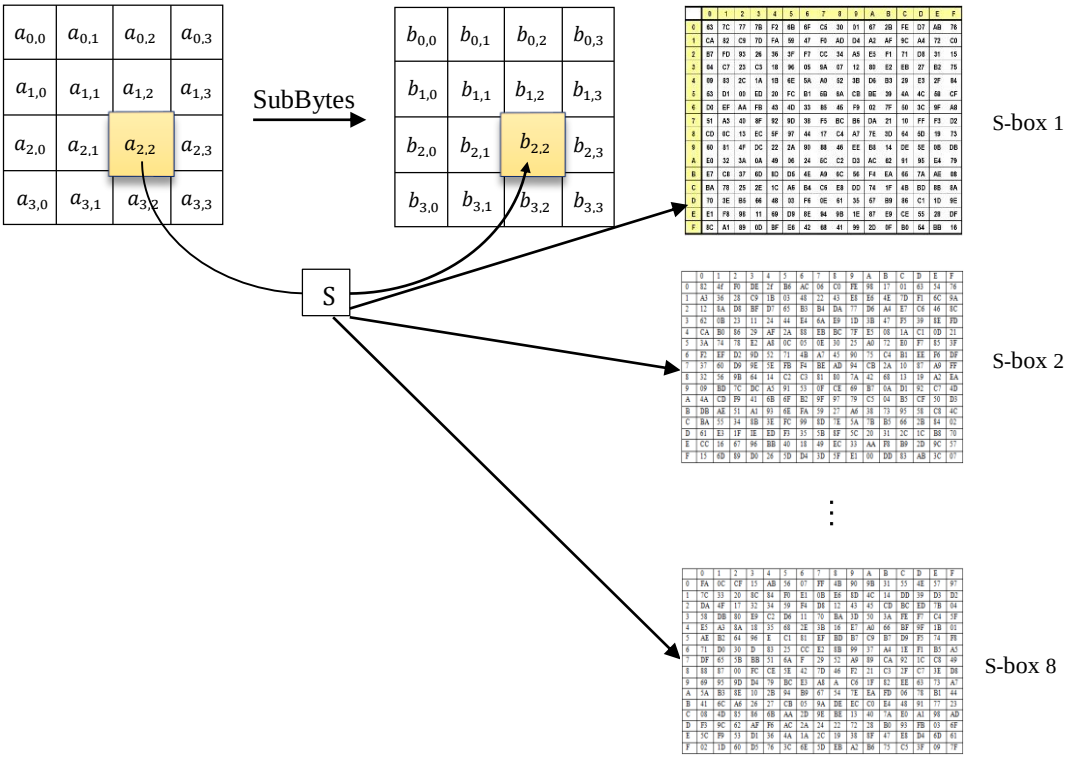


Fig. 7 Multi S-Box process

Algorithm 1: The generation of multi S-Box process [3].

***Input**: Randomly (8) values as { Rndm_Key[k], Cons_c[k] , k=1,2,……..,8 }*
***Output**: Diverse (8) S-Boxes { (S-box[m][n])k ; (S-1-box[m][n])k ; k=1,2,…..,8 }*
Steps: ***1.** Choice 8 keys Rndm_Key[k] // to create a unique S- box when each key has the inverse to recreate Inverse S-box* ***2.** Choice Eight different values Cons_c[k]* ***3.** For each key Rndm_Key[k] & relates constant Cons_c[k] generate its own S-box[m][n]* *(S-box[m][n])k= Rndm_Key[k] * mulp[r][c] + Cons_c[k]* *Where mulp[r][c] represent the multiplicative inverse in GF(28)* ***4.** Use (S-box[m][n])k in encryption operation.*

Algorithm 2: The Sub-Byte transformation process [3].

Input *:plain text Block { State[Row][Colum]}; r,c={1,2,3,4}* *8 S-Box{(S-box[m][n])k ; (S-1-box[m][n])k ; k=1,2,.....,8 .* *matrix of Key sharing {Key-Enc\Dec[4][4]}*
Output: *cipher text Block {State[Row][Colum]} ; r , c ={1,2,3,4}*
Steps: ***using new S- box[m][n] to encrypt each block :*** ***1.*** *For Each Row in State matrix, Do* ***2.*** *For Each Colum in State matrix, Do* ***3.*** *Y = (Stat [Row][Colum])&0x0f;* *X = (Stat [Row][Colum]>>4)&0x0f;* *X, Y = the index of row and colum in S-Box* ***4.*** *State[Row][Colum] encrypt by using the index of each* *S-Box ,; Key_Enc[4][4] : State[Row][Colum]=(S-box[x][y])*

Steps in Algorithm 2 add much more complexity because each key (Rndm_key and Cons_c) needed for generating the S-Boxes consist of 128 bits; this leads to a number of possible generated s-boxes using Eq.1.

$$complexity\ (key\ space) = 2^{128} * 2^{128} = 2^{256} \qquad (1)$$

In addition, there is a random distribution of the S-Boxes which adds more complexity. The use of another key leads to much more complexity as:

$$random\ distribution\ \ sbox = 8! \qquad (1a)$$

The total complexity (i.e. new key space) for this algorithm will therefore be of a high level, adding more security for this operation. This means that it is very difficult to find which S-Box is used in the encryption, thus adding more resistance.

3.3.2 Implementation of MixColumn Function

In AES algorithm the MixColumn function is the third function in the AES algorithm which accounts as the most computationally expensive operation, where the input matrix is multiplied over GF(2^8) [3]. The key matrix used in forward and inverse MixColumn transformation functions, that operate on the state matrix, is single and with fixed dimension [4*4]. The scheme in [3] tends to improve MixColumn transformation by splitting the key matrix into four parts, each part representing a different key with dimension [2*2]. The state matrix is also divided into four parts with [2*2] dimension, each part corresponding to one of the keys into a similar position in the key matrix. This method will be used in this research to develop new technique which helps to suggest the lightweight solution.

In the product matrix and in each part, any element is the sum of the products of the elements of one row and one column. In this case, the individual additions and multiplications are executed in GF(2) and GF(2^8). The transformation can be defined by the following matrix multiplication between the State and key matrices. Details of the algorithm to improve MixColumn

transformation is shown in Fig. 8. It is assumed that all four parts of the keys are fixed. The reason is to reduce the keys exchanging operation between the sender and receiver which causes more load in the network. Especially in wireless sensor networks and MANET when the nodes enter and leave frequently, this causes a repetition of the rekeying operation between them.

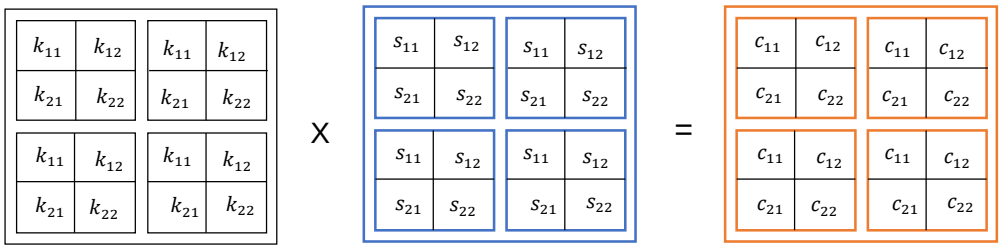


Fig. 8 Process of Multiplication in the Proposed MixColum Function [3]

So the operation to multiply two matrixes of [4*4] dimension, when they divided in 4 parts each part with 2*2 dimension:

$$C11 = (K11 * S11) + (K12 * S21) \quad ,$$
$$C12 = (K11 * S12) + (K12 * S22) \quad ,$$
$$C21 = (K21 * S11) + (K22 * S21) \quad ,$$
$$C22 = (K21 * S12) + (K22 * S22) \quad ,$$

Where: k represents the key, S represent the state, and C represent the cipher. This process will be applied on each part of the matrix.

From the above equations, the total number of mathematic operations can be calculated as follows: each element in the matrix required three operations, two multiplications and one summation. So the total number of operations for whole matrix is given as follow:

16 * 3 = 48 (mathematic operation)

This method will reduce the number of operations needed to multiply two matrices because it reduces the summation and multiplication process, consequently leading to the reduction of the time and energy consumption needed for this function, as shown in Algorithm 3. Fig 9 illustrates the new design steps of the MixColumn function.

Algorithm 3: the MixColumn function [3]

Input *:plain text Block { State[Row][Colum] :; r,c=1,2,3,4} and The mix colum k-encryption Mix_Key[r][c], r,c = 1,…….,4*
Output*: Ciphertext text Block C_Blok[r][c], r,c =1,……,4 .*
Steps: *1: Divided plain Block[r][c] and Mix_Key[r][c] into 4 portions each one of size 2*2, for each portion of Mix_Key[2][2] calculate its inverse matrix.*   *2: Multiplication of each portion of P_Block[2][2]* Mix_Key[2][2] to produce C_Matrix[2][2].*   *3: Rebuild the C_Block[r][c] r,c=1,..,4, from each portion of C_matrix[2][2]*

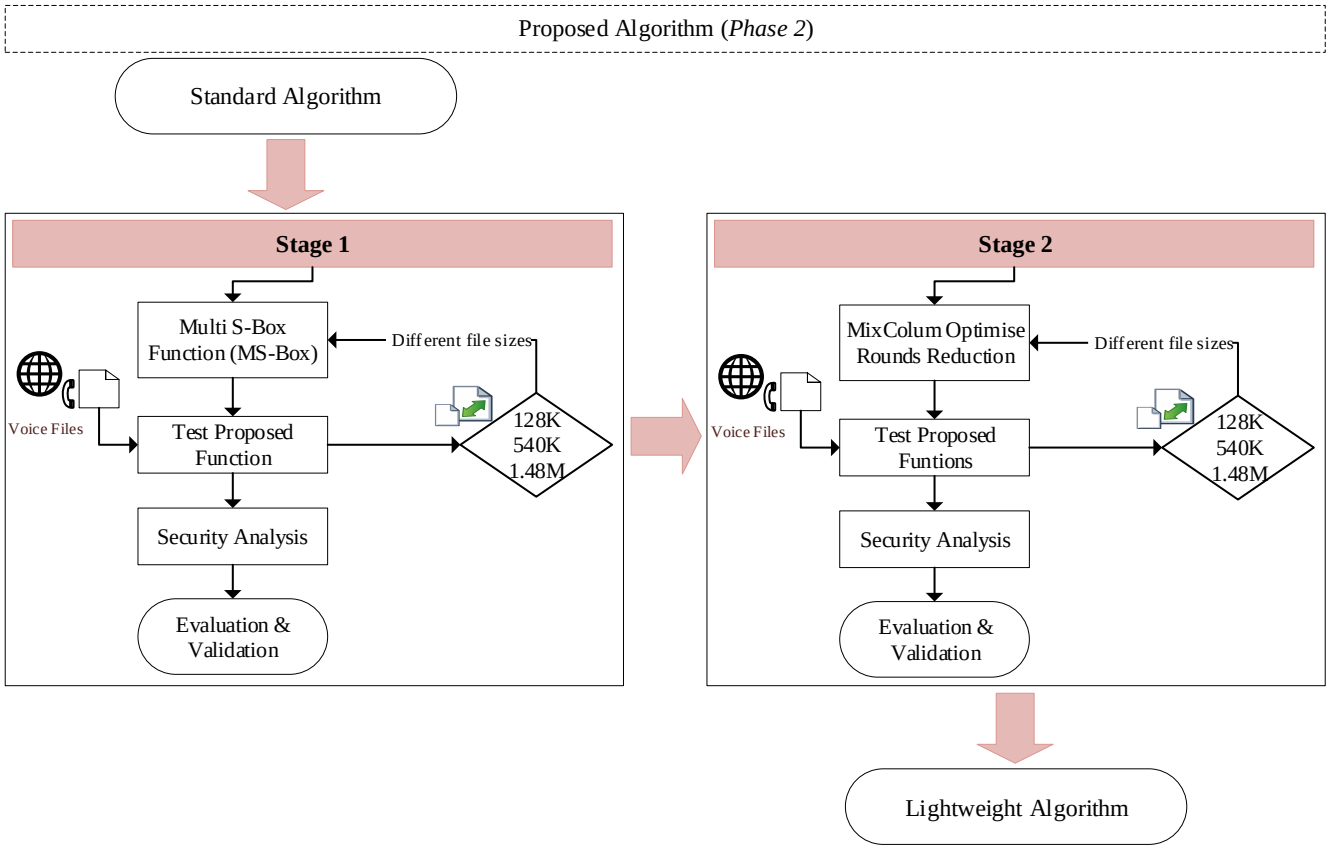


Fig. 9 Proposed Design (phase 2)

3.3.3 Proposed New Nine-Rounds Iteration

The high complexity that has been achieved in section (3.3.1) for Sub-Byte transformation, allowed the reduction of the number of rounds in the AES algorithm, making it more suitable for MANET, which has nodes with power limitation. Therefore, in this section, the aim is to decrease the power consumption and execution time and keep the security and complexity at the same level.

Reducing the number of rounds in the new algorithm processes will decrease the power consumption and at the same time reduces the execution time and keeps the complexity at the same because the complexity was already increased in the S-Box stage as mentioned in section (3.3.1). According to [15], five rounds of AES is still critical in term of security. While the last successful attack was on seven rounds [4]. Therefore, nine rounds are proposed for the iteration algorithm, to be more secure and resist different attack. The mathematical model of the new proposed nine round is illustrated below:

$$P = power,\ T = time\ to\ execute\ the\ encryption\ process$$

$$P(Enc), power\ consumed\ for\ encryption\ process = \sum P(round), Power\ consumed\ for\ each\ encryption\ round$$

$$T(Enc) = \sum T(round)$$

So, by reducing the number of rounds to 9, the power of the new algorithm is given in Eq. 2.

$$New\ algorithm\ power = 0.9 * P(AES) \quad (2)$$

$$New\ algorithm\ time = 0.9 * T(AES)$$

The proposed 9-rounds algorithm is capable to decrease the power consumption (compared with AES) by up to 10% and reduce the execution time by nearly the same proportion (results are demonstrated in the result section), while keeping the security at the same level, because the complexity of the algorithm is already addressed during the S-Box generation stage and not the round stage. Algorithm 4 illustrates the proposed 9-rounds algorithm.

Algorithm 4. Proposed 9-rounds steps

***Input**: plaintext Block {State [Row][Col]; (Row, Col=1,2,3,4)}.* *With 1-8 S-boxes, generated in algorithm 1.*
***Output**: cipher text C {[Row][Colum]; (Row Colum=1,..,4)}*
Steps: ***Encrypt using nine rounds*** *For (i = 0; i<8; i++)* *SubBytes(); // subbytes transformation proposed in algorithm (2)* *ShiftRows();* *MixColumns(); // mixcol proposed in algorithm (3)* *AddRoundKey();* *Last round;* *End.*

3.3.4 The New Overall Design Framework

The new design adds new features to the functions of the AES algorithm as described earlier, which results in a reliable trade-off between security and QoS parameters. The power and time consumption is reduced as well as the security of the algorithm is increased. The new overall key space of the new algorithm is given in Eq. 3.

$$Dual\ Key\ (\ new\ key\ space) :\ \ 2^{128} * 2^{256} * 8! \qquad (3)$$

Attackers need to know two keys at the same time and it is very difficult to guess all of them together; so, even if one key is hacked, it would be more complicated to find the other key. Also, theoretically, if an attacker uses the Brute-force attack to find all the keys, then a huge time will be needed for it to be successful, greater than 5.3*1023 years. Therefore, in this case the security level for this algorithm is maintained or even increased.

As mentioned earlier, the new algorithm is more suitable for a wireless environment because of the new features. The following section provides more analysis linked back to these features. Fig. 10 demonstrates the proposed enhancement in the new AES algorithm.

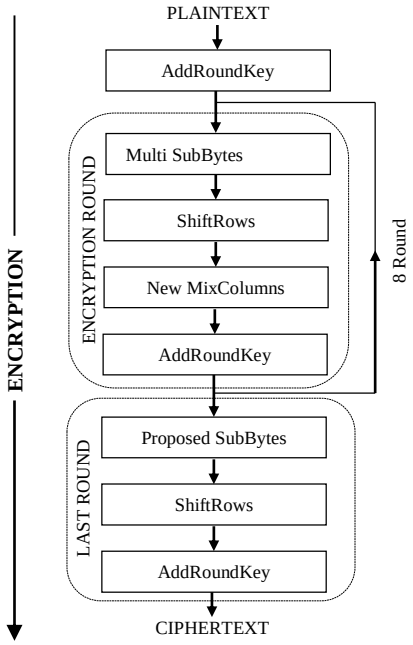


Fig. 10 The Proposed Encryption Algorithm

To validate and evaluate the new algorithm, a comparison with the standard AES algorithm has been carried out to compare the security strength parameters such as the randomness and frequency parameters. These parameters can be measured by conducting several kinds of tests like binary histogram, floating frequency, poker test and brute-force attack, all these parameters are explained in the security analysis subsection. Also, time and power consumption are compared. The algorithm's power consumption can be measured by calculating the total of computing cycles which are used in processes related to cryptographic tasks. For the calculation of the total energy and cost of encryption, the following equations are used similar with those defined in [10].

$$B\ cost\_encry\ (ampere - cycle) = \tau * I \quad (4)$$

$$T\ energy\_cost\ (ampere - seconds) = \frac{\boldsymbol{B}\ cost\_encry\ (ampere - cycle)}{F\ (cycle/sec)}$$

$$E\ cost\ (Joule) = T\ energy\_cost\ (ampere - seconds) * V \quad (5)$$

Where $B\ cost\_encry$ is a basic cost of the encryption process (ampere-cycle), τ is the total number of clock cycles, I is the average current drawn by each CPU clock cycle, $T\ energy\_cost$ is the total energy cost (ampere-seconds), F is the clock frequency (cycles/sec), and $E\ cost$ (Joule) is the energy cost (i.e. energy consumed).

By using the cycles, the operating voltage of the CPU, the average current drawn for each cycle, and the energy consumption of cryptographic functions can be calculated. For example, on average, each cycle consumes roughly 270 mA on an Intel 486DX2 processor or 180 mA on Intel Strong.

4. Experiment and results

The testing has been conducted in a laboratory environment using a wireless laptop running Visual Studio 2015 using Console app to avoid any load on the processor. The proposed algorithm has been tested on an audio file with different sizes. After that, the security and performance analysis has conducted to measure the new algorithm strength. Security analysis has been conducted to prove the algorithm strength and its validity to suit and secure wireless networks. The program code will call the file and open and read it. Then execute the encryption process by reading and encrypt block by block, each block has 16 byte. The proposed algorithm uses dual keys; the first key is a set of multi values up to 16 elements. Each value in the key set has another value related to it, as in AES Rijndael algorithm leading to build different S-boxes with its related inverse S-Box. The key below is used to generate multi S-Box tables before the encryption process start:

$$Rndom\_Key[8] = \{0x67, 0x85, 0x25, 0xb5, 0xA4, 0xf1, 0x19, 0x4c\}$$

$$Cons\_c[8] = \{0x82, 0x45, 0xc4, 0xa5, 0x7b, 0x63, 0xd5, 0xc1\}$$

Represent the first key, based on hexadecimal, each value in the key with its related constant (cons_c) value, can create unique S-Box.

After executing the encryption program (source.cp) by the software, the command line appears and ask to enter the main encryption key. The key used in this experiment is *123456789abcdef123456789abcdef12*.

The length of the key is 32 char (i.e. 16 byte). And for more accuracy, the code has been run for five times and the average output has been calculated.

4.1 Test Results

The output file, as a hexadecimal representation of the voice, is fully encrypted and it is not understandable. Fig. 11 below shows the output data which is clearly random and different. During to lessening to the encrypted file, it was completely unclear and nobody can understand it, so this means that the encryption was successful. In this section we are going to show and analyse in details, all the performance and security metrics, to prove that research objectives were met.

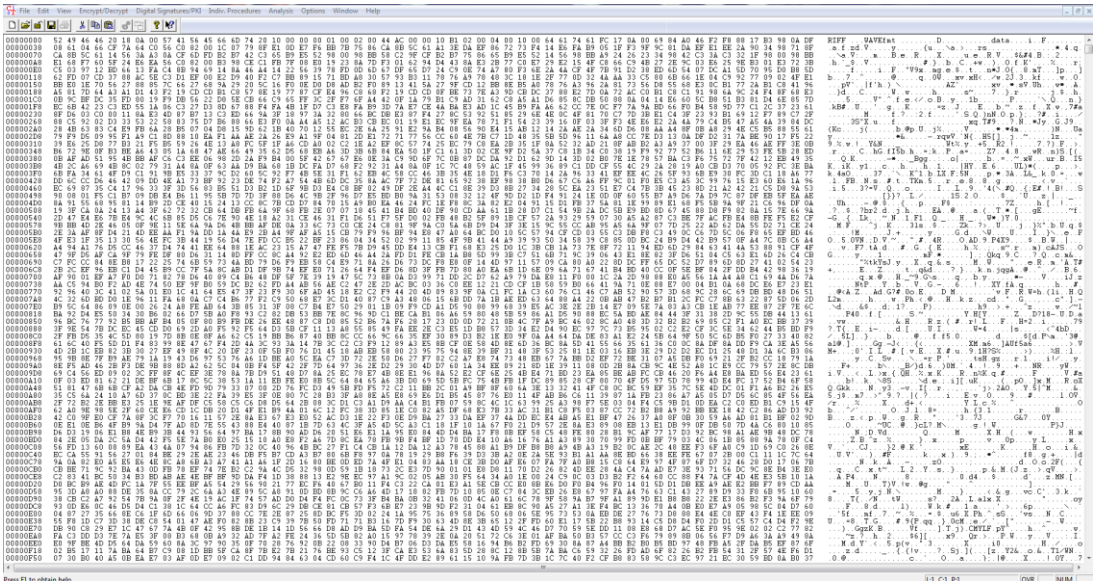


Fig. 11 Snapshot of the Encrypted Voice File

In this experiment, both standard and proposed algorithm has been tested. Table 1 and Table 2 show the time and Energy consumed by both the standard AES and the proposed algorithm for the encryption and decryption process.

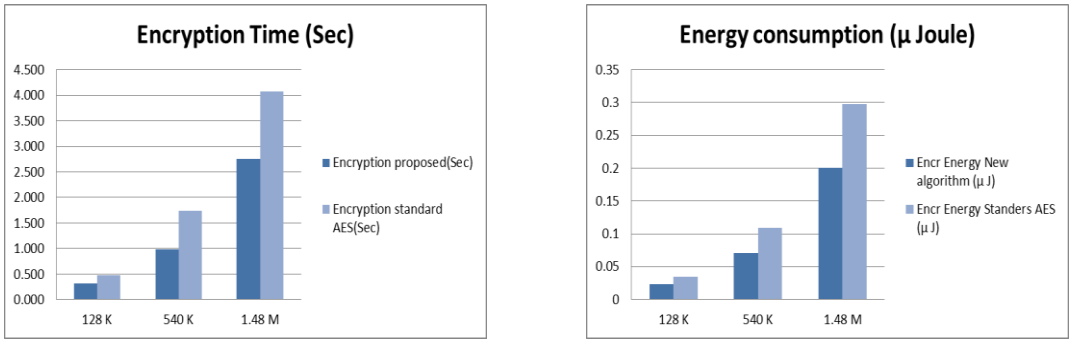
Table 1 Comparison of the execution time for the standard AES and the proposed algorithm

File Size	Encryption Time Sec (Standard AES)	Decryption Time Sec (Standard AES)	Encryption Time Sec (Proposed Algorithm)	Decryption Time Sec (Proposed Algorithm)
128 K	0.4768	0.462	0.314	0.324
540 K	1.742	1.387	0.977	1.086
1.48 M	4.0776	3.885	2.747	2.833

Table 2 Comparison of the Energy consumption for the standard AES and the proposed algorithm

File Size	Encryption Energy µJ (Standard AES)	Decryption Energy µJ (Standard AES)	Encryption Energy µJ (Proposed Algorithm)	Decryption Energy µJ (Proposed Algorithm)
128 K	0.035	0.034	0.023	0.024
540 K	0.109	0.101	0.071	0.079
1.48 M	0.298	0.284	0.201	0.207

The results of the proposed algorithm show significant improvements in both the encryption and decryption process. More than 33% improvement is achieved in the execution time for different file sizes. For example, for the file sized 1.48M, an improvement of roughly 34% is achieved in the encryption time by the proposed algorithm. The same behaviour is exhibited in the energy consumption metric in which more than 33% enhancement is achieved by the proposed algorithm (see fig.12), compared with the standard AES. This provides evidence that the proposed MixColumn enhancement and 9-round iteration reduce the executing cost of the encryption operation. All these new features can be noted in fig.12 below.



(a) (b)

Fig. 12: Comparison of (a) encryption time, (b) power consumption, of the standard AES and the proposed algorithm

Table 3 shows the comparison of features between the standard AES and the proposed algorithm. It can be seen that there are many important differences between them. Also, it is clear that the file size is still the same, which will therefore not affect the memory size and the bandwidth of network paths.

Table 3 The comparison of features between the standard AES and the proposed algorithm.

Property	AES	New Algorithm
Security	High	High
S_box	Single	Multi
Num. of keys	1	2
Key space	2^{128}	$2^{128}*2^{256}*8!$
Binary Histogram	Random	Random
Poker Test	Pass	Pass
Block length	16 Bytes	same
No. Rounds	10	9
Encryption time	**T**	**0.65 T**
Power consumption	**P**	**0.65 P**
Output = input size	Yes	Yes

The proposed algorithm has a high level of security because the complexity of finding the keys is increased, after adding more keys. Each key has 2^{128} of complexity, as mentioned in Table 6, in addition to using eight S_Boxes which increased the

complexity by 8!; this complexity was multiplied by the number of rounds, 10 rounds in the standard AES and 9 rounds in the proposed change. Meanwhile, the execution time and power consumption have decreased by 35% and this is because of reducing the rounds and the new MixColumn function. To validate and assess the security strength of the proposed algorithm, the next section evaluates in detail the security metrics of the proposed algorithm and compares them with those of the standard AES.

4.2 Security Analysis

Appropriate security parameters are required to investigate the degree of randomness and encryption quality of the output file (binary sequences) produced by the proposed algorithm, statistical testing and mathematical measurements [50]. These metrics could then be used to collect evidence about the output sequences and to check whether they are truly random and have a high encryption quality, and consequently proof the algorithm's suitability and capability to be used safely in the converged network applications [51]. The aim of the security analysis used in this research is to confirm the security strength of the proposed schemes as this can be considered as one of the contributions of the paper as there is a lack of security analysis in many research works. According to [50], the security metrics are essential to measuring the security strength of any security system. A security analysis is conducted to test the security level of the proposed new algorithm. Many security parameters have been measured, such as Entropy, Binary Histogram and Floating frequency test. The randomness of the encrypted data, including poker test and frequency test are carried out as well. CrypTools 1.4 for cryptography and cryptanalysis [15] has been used to carry out these tests. All the tests have been conducted on an encrypted audio file format (.wav) of 540K size as a sample.

4.2.1 Entropy Analysis

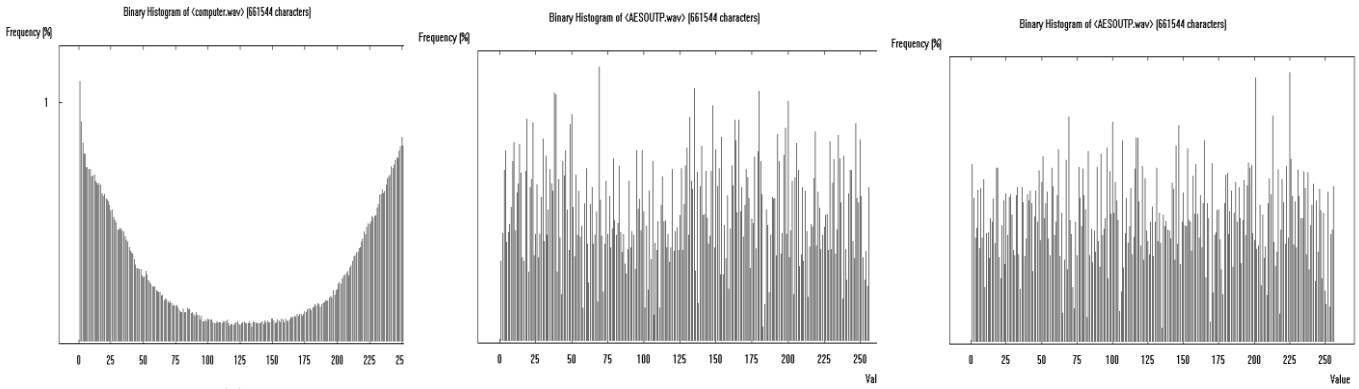
The entropy of a document is an index of its information content. The entropy is measured in bits per character. Table 4 shows the Entropy test result for the encrypted file in both the standard AES and the proposed algorithm. The proposed algorithm achieves the same score compared to the standard algorithm, which was 7.99 of the maximum possible value of 8. This means that there is still a same security level in the proposed algorithm.

Table 4. ENTROPY ANALYSIS

Audio file size / Byte	before	AES algorithm			New algorithm		
	Entropy	Entropy	Max. possible Entropy	Possible byte value	Entropy	Max. possible Entropy	Possible byte value
540k	7.79	7.99	8	256	7.99	8	256

4.2.2 Binary Histogram

Binary histogram expresses the frequency distribution of the characters of the document in graphical form. The figures below show the security analysis for the encrypted file for both the standard AES and the proposed algorithm. Fig. 13a represents the Binary Histogram of the original audio file before the encryption



(a) (b) (c)

Fig. 13: (a) Binary Histogram of the audio file before encryption,
(b) Binary Histogram of the audio file after encryption by the proposed algorithm,
(c) Binary Histogram of the audio file after encryption by standard AES

Fig. 13b illustrates the Binary Histogram of the encrypted file using the proposed algorithm. It is clear that there is a considerable difference between the original file and the output of the proposed algorithm, implying a strong security level. Fig. 13c shows the Binary Histogram of the encrypted file for the standard AES, in comparison with the Binary Histogram presented for the new algorithm in Fig. b. and this mean that there is no clear effect on the algorithm strength and the security level has been maintained.

The following table describe the previous diagrams and illustrate their values. It shows the frequency for some characters for the plain and cipher which encrypted by standard AES and the proposed Lightweight Encryption Algorithm (LEA).

Table 5. BINARY HISTOGRAM

Character value (Dec.)	Equivalent value (Hex)	Frequency (Plain Text) (%)	Frequency Cipher (LEA) (%)	Frequency Cipher (AES) (%)
1	01	**1**	0.68	0.61
25	19	**0.49**	0.67	0.51
50	32	**0.25**	0.33	0.55
75	4B	**0.13**	0.47	0.31
100	64	**0.09**	0.62	0.75
125	7D	**0.06**	0.50	0.35
126	7E	**0.061**	0.70	0.31
127	7F	**0.062**	0.61	0.41
128	80	**0.063**	0.82	0.46
150	96	**0.12**	0.57	0.52
175	AF	**0.13**	0.61	0.47

200	C8	**0.22**	0.42	0.37
225	E1	**0.46**	0.62	0.93
250	FA	**0.77**	0.40	0.18

Frequency means the number of times repeated in the text. From the above table it is clear that a good confusion has been achieved in the cipher for both TKE and AES, For example, the character (125) in the diagram, (125 Dec.= 7D Hex= } ASCII), has 0.5% frequency, While actual frequency in the plain file is 0.6%. Also, 75 has 0.47% While actual frequency in the plain file is 0.13%. So, all the frequency values have big differences from the plain file. This adding more confusion to the cipher and will confuse the attacker making it more secure and high resistance to cryptanalysis. To calculate the differences between the Plain file and cipher mathematically, the following equation computes the number of each character in the file:

$$no.of\ char = Total\ no.of\ char * \frac{char\ freq.}{100}$$

For example, the number of character repeating of (125≠7D) =

$$no.of\ 7D = 661544 * \frac{0.5}{100} = 3307$$

While in the Plain file:

$$no.of\ 7D = 661544 * \frac{0.06}{100} = 397$$

Another example for (150≠96):

$$no.of\ 96 = 661544 * \frac{0.57}{100} = 3770$$

In Plain file:

$$no.of\ 96 = 661544 * \frac{0.12}{100} = 794$$

Thus, by using the proposed cryptosystem, the cipher will not supply any useful information related to the plain-file. And because of there are significant differences between the plain and cipher file, the statistical attacks will be infeasible.

4.2.3 Floating Frequency

Floating frequency is a characteristic of local information content at individual points in the document. The floating frequency specifies how many different characters are to be found in any given 64-character long segment of the document. Fig. 14 show the floating frequency parameter for both the original audio file, and the encrypted file for the proposed algorithm.

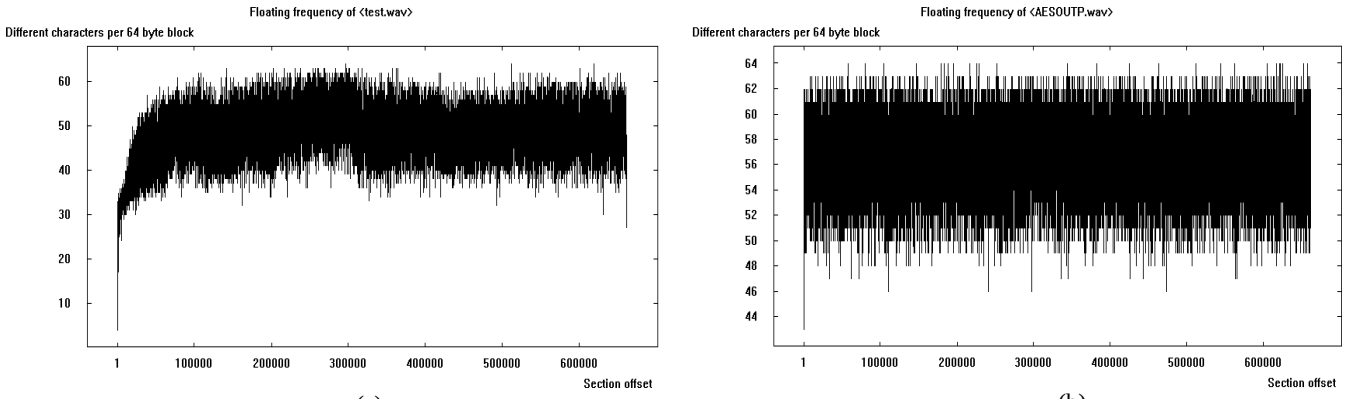


Fig. 14: (a) Floating Frequency of audio file before encryption
(b) Floating Frequency of the audio file after encryption by new algorithm

It is clear from the fig. 14a above that the floating frequency for the encrypted file is at a good level in which most of the frequency is between 50 - 60 different characters per 64-byte block compare with fig.14b where the frequency ranges between 10 – 55. This means that there is a significant crypto strength in the proposed algorithm, keeping the diffusion in the cipher and leading to more complexity in the relationship between the cipher and the plaintext. All the figures above and the analysis carried out demonstrate that considerable security level has been achieved through the proposed encryption algorithm and it maintained the complexity of the algorithm compared with standard AES.

4.2.4 Poker Test

In addition to the previous analysis, some tests have been carried out, such as poker test, to test the randomness of output audio file. These tests have been carried out on the encrypted audio file and both of these tests were passed, as shown in Fig 15.

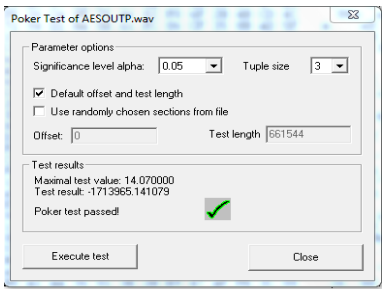


Fig.15. Poker test result

4.2.5 Brute-Force attack

A Brute-Force attack was tested on the 128-bit key. It shows that a huge time was taken to analyse the key. Performance of this attack is shown in fig. 16. This attack will never reveal the keys because it will take a massive time.

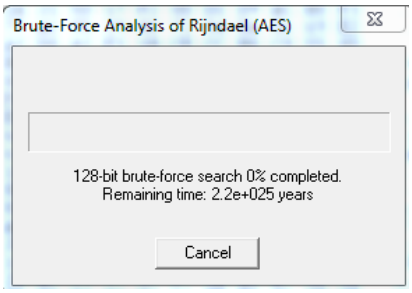


Fig.16. Brute-Force attack

From all above analysis and tests, it is clear that the proposed algorithm achieves a significant security level and it has a good resistance to different attacks such as brute-force, chosen cipher text and cipher-text only attack, because of the confusion and the diffusion characteristics it has. Also, the high randomness demonstrated in the above subsections shows that the suggestion of 9-rounds iteration, did not affect the security level. Security level analysis parameters including binary Histogram, Floating frequency and randomness are measured and compared with the standard AES, which show the same security strength.

Table 6 presents a comparison of our work with published works in [15] and [3]. It compares the power saving percentage and the security level against the standard AES.

Table 6. COMPARISON

Work ref.	**Power saving (%)**	**Overall Security against AES**
[15]	40 %	low
[3]	0 %	high
Proposed algorithm	35 %	Same/high

From the table above, it can be seen that a significant trade-off solution between security and QoS metrics has been achieved in the proposed algorithm, making it more suitable for real time traffic over wireless networks and devices with limited resources.

5. Conclusions and future work

A new encryption algorithm for real time traffic over wireless networks is proposed, tested and validated in this paper. Using robust changes in the standard AES functions, the proposed algorithm can provide a reasonable level of security and meets the performance of such networks and devices. The change in the Sub-Byte function is the increase of the degree of complexity within the same delay time during the encryption and decryption processes. The change in the MixColumn function reduces the number of operations needed to multiply two matrices as it reduces the summation and multiplication process, consequently leading to the reduction of the time and energy consumption needed for this function. Finally, the proposed 9-round change is capable to decrease the power consumption by up to 10% and reduce the execution time by nearly the same proportion. The main contribution of the paper is reducing the execution time and power consumption of the proposed new algorithm, besides, maintaining/increasing the security level comparing with the standard AES algorithm. A comprehensive security analysis is conducted to test the validity of the proposed algorithm in terms of the high complexity and randomness of the proposed algorithm which can resist different types of attacks. The proposed algorithm is suitable for wireless devices with limited

resources and it achieves a considerable trade-off solution between security and QoS, thus it exhibits its applicability for any wireless networks where the resources are limited.

Future directions to further develop the algorithm includes increasing the Sub-Byte function complexity and adding third key in the algorithm to increase the security level, depending on the resources and limitations of the network.

Conflict of Interest:

The authors declare that they have no conflict of interest.

References

[1] W. Hu and Cao, "Quality-Aware Traffic Offloading in Wireless Networks," *IEEE TRANSACTIONS ON MOBILE COMPUTING*, vol. 16, no. 11, March 2017.

[2] B. Mohd, T. Hayajneh, A. Vasilakos "A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues" *Journal of Network and Computer Applications*, vol. 58, pp. 73-93, 2015.

[3] N. Ali, A. Rahma, S. Yousef and M. Jaber, "Random Key Permutation Stream Algorithm Based on Modified Functions in AES Algorithm," International Journal of Engineering and Technology, vol. 4, no. 6, pp. 367 - 373, June 2014.

[4] B. Bahrak and M. R. Aref, "Impossible differential attack on seven-round AES-128," *IET Information Security, Institution of Engineering and Technology*, vol. 2, no. 2, pp. 28-32. [avalible online on]: https://ieeexplore.ieee.org/document/4558840/, July 2008.

[5] L. S. Abhiram, B. K. Sriroop and H. L. Punith.Kumar , "FPGA implementation of dual key based AES encryption with key Based S- Box generation," India, 2015.

[6] Stallings, W., Cryptography and Network Security: Principles and Practice (7th Edition). 7 th ed. Harlow- England: Pearson Education Limited, 2017.

[7] S. Kalra and S. Sood, "Advanced password based authentication scheme for wireless sensor networks," *Journal of Information Security and Applications*, vol. 20, pp 37-46, February 2015.

[8] D. Salama and M. Hadhoud, "Evaluating the effect of symmetric Algorithms on Power consumption for Different Data types," *International Journal of Network Security*, vol. 11, no. 2, pp. 78-87, 2010.

[9] J. Wang, Q. Gao, P. Cheng, Y. Yu, "Lightweight Robust Device-Free Localization in Wireless Networks," *IEEE Transactions on Industrial Electronics, IES*, vol. 61, no. 10, 2014.

[10] W. Jiehong and I. Detchenkov, "A Study on the Power Consumption of Using Cryptography Algorithms in Mobile Devices," China, 2016.

[11] H. Albonda, S. Tapaswi, S. Yousef and M. Cole, "The impact of mobility and node capacity on voice traffic," *International Journal of System Assurance Engineering and Management*, vol. 8, no. 33, pp. 1 - 9, March 2017.

[12] Ayyappadas, Devassy, S. George and A. Devassy, "Survey of Symmetric Cryptographic Algorithms," *Journal of Electronics and Communication Engineering (IOSR-JECE)*, pp. 65-75, 2014.

[13] s. Cheng, P. Chen and C. Lin, "Traffic-Aware Patching for Cyber Security in Mobile IoT," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 29 - 35, 2017.

[14] S. Morgan, "Cyber Security Cost," Forbes, available online on: https://www.forbes.com/sites/stevemorgan, US, 2017.

[15] A. Msolli, A. Helali and H. Maaref, "Image encryption with the AES algorithm in wireless sensor network," Tunisia, July 2016.

[16] A. S. Rahma and B. Z. Yaco , "Real-Time Partial Encryption of Digital Video using Symmetric Dynamic Dual Keys Algorithm," *Engineering and Technology Journal*, vol. 30, no. 5, pp. 710-728, 2012.

[17] R. Chandramouli, S. Bapatla and K. P. Subbalakshmi, "Battery power-aware Encryption," *ACM Transactions in Information and System Security*, vol. 9, no. 2, pp. 162-180, 2006.

[18] Y. Zou, J. Zhu, X. Wang and L. Hanzo, "A Survey on Wireless Security: Technical Challenges, Recent Advances, and Future Trends," in Proceedings of the IEEE, vol. 104, no. 9, pp. 1727-1765, Sept. 2016.

[19] Emmanouis and Christos, "Security model for emergency real-time communications in autonomous networks," *Springer- Information Systems Frontiers, A Journal of Research and Innovation*, vol. 14, no. 3,pp.541–553[online]:https://link.springer.com/article/10.1007/s10796-010-9259-8, 2012.

[20] F. Hazzaa, S. Yousef, E. Sanchez and M. Cirstea, "Lightweight and Low-Energy Encryption Scheme for Voice over Wireless Devices," *IECON 2018 - 44th Annual Conference of the IEEE Industrial Electronics Society*, Washington, DC, 2018, pp. 2992-2997.

[21] M. Nagendra and M. C. Sekhar, "Performance Improvement of Advanced Encryption Algorithm using Parallel Computing," *International Journal of Software Engineering and Its Applications*, vol. 8, no. 2, pp. 287-296. [availble online]: https://pdfs.semanticscholar.org/. 2014.

[22] Alamsyah, A. Bejo and T. Adji, "AES S-box construction using different irreducible polynomial and constant 8-bit vector," Taiwan, 2017.

[23] Sahu, S. K. & Kushwaha, A., 2014. Performance analysis of Symmetric Encryption algorithm for Mobile ad hoc networks. International Journal of Emerging Technology and Advanced Engineering, 4(6), pp. 619-624.

[24] A. Prakash, M. Satish, T. Sai and M. G., "Improving Cloud Security Using Multi Level Encryption and Authentication," *International Journal of Innovative Research in Information Security (IJIRIS)*, vol. 2, no. 8, pp. 1-8, Aug. 2015.

[25] Liang and Han, "Multimedia Traffic Security Architecture for the Internet of Things," *IEEE Network*, pp. 35-40, 2011.

[26] I. Das , S. Nath , S. Roy and S. Mondal, "Random S-Box generation in AES by changing irreducible polynomial," India, 2013.

[27] G. RAMESH and R. UMARANI, "Performance Analysis of Most Common Symmetrical Encryption Algorithms," *International Journal of Power Control Signal and Computation(IJPCSC)*, vol. 13, no. 1, pp. 42-45, 2012.

[28] Behnam Dezfouli, I. Amirtharaj, C. Li, "An energy measurement platform for wireless IoT devices," *Journal of Network and Computer Applications*, vol. 121, pp. 135-148, 2018.

[29] F. Hazzaa and S. Yousef, "Performance Analysis for Traffics in Mobile Ad Hoc Network," in *11th International Conference on Global Security, Safety & Sustainability ICGS3 - Springer International Publishing AG*, London, 2017.

[30] A. Popov, "Prohibiting RC4 Cipher," Internet Engineering Task Force, WA, USA, 2015.

[31] P. Gope, T. Hwang, " A Realistic Lightweight Anonymous Authentication Protocol for Securing Real-Time Application in WSN," *IEEE Transactions on Industrial Electronics, Industrial Electronics Society*, vol. 63, no. 11, pp. 7124-7132, 2016.

[32] M. Masoumi and M. H. Rezayati, "Novel Approach to Protect Advanced Encryption Standard Algorithm Implementation Against Differential Electromagnetic and Power Analysis," *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, vol. 10, no. 2, pp. 256-264. [available online]: (www.ieeexplore.ieee.org), Digital Library, February 2015.

[33] G. Bansod, N. Raval and N. Pisharo, "Implementation of a New Lightweight Encryption Design for Embedded Security," *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, vol. 10, no. 1, pp. 142-151, 2015.

[43] C. Lee, "Biclique cryptanalysis of PRESENT-80 and PRESENT-128," *The Journal of Supercomputing*, vol. 70, no. 1, pp. 95-103 doi:10.1007/s11227-014-1103-3. ISSN 0920-8542., 2014.

[35] S. Faghihi , M. Hossein and M. Dakhilalian, "Biclique cryptanalysis of MIBS-80 and PRESENT-80 block ciphers," *Security and Communication Networks*, vol. 9, no. 1, pp. 27-33, 2015.

[36] P. Zhang, C. Lin and Yixin, "A Lightweight Encryption Scheme for Network-Coded Mobile Ad Hoc Networks," *IEEE TRANSACTIONS ON PARALLEL AND DISTRIBUTED SYSTEMS*, vol. 25, no. 9, pp. 2211-2220, 2014.

[37] Liang and H.-C. Chao, "Multimedia Traffic Security Architecture for the Internet of Things," *IEEE*, no. 0890-8044/11/, 2011.

[38] W. Stallings, Cryptography and Network Security Principles and Practice, 5th ed., England: Pearson, 2012.

[39] J. G. Matesanz, A. L. Orozco and L. J. Villalba, "Security Issues in Mobile Ad Hoc Networks," International Journal of Distributed Sensor Networks, vol. 2012, no. 1, pp. 1-6., Nov. 2012.

[40] R. D. Pietro, S. Guarino and Verde, "Security in wireless ad-hoc networks," *Elsevier- Computer Communications*, vol. 51, pp. 1-20 [available at ScienceDirect], 2014.

[41] s. Cheng, P. Chen and C. Lin, "Traffic-Aware Patching for Cyber Security in Mobile IoT," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 29 - 35, 2017.

[42] D. Salama and M. Hadhoud, "Evaluating the effect of symmetric Algorithms on Power consumption for Different Data types," *International Journal of Network Security*, vol. 11, no. 2, pp. 78-87, 2010.

[43] S. S. Kolahi, K. Mudaliar and Z. Gu, "Impact of IPSec security on VoIP in different environments," Milan,Italy, July,2017.

[44] F. Hamada and Rahman, "Impact of IPSec on MANET," China, 2016.

[45] L. Zhou, "Distributed Scheduling Scheme for Video Streaming over Multi-Channel Multi-Radio Multi-Hop Wireless Networks," *IEEE JSAC*, vol. 28, p. pp. 409–19, 2010.

[46] H. V. Zhao, W. S. Lin and K. Liu, "A Case Study in Multimedia Fingerprinting: Behaviour Modelling and Forensics for Multimedia Social Networks," *IEEE Sig. Proc. Mag*, vol. 26, no. 1, p. 118–39, Jan. 2009.

[47] P. Singh, "AES Keys and Round Functions for Data Security," *International Journal of Computer Applications (0975 – 8887)*, vol. 39, no. 11, pp. 23-27, Fabruary 2012.

[48] B. Trapnell and C. French, "Cryptographic Module Validation Program," National Institute of Standards and Technology NEST, USA, 2016.

[49] F. Hazzaa, S. Yousef, N. H. Ali and E. Sanchez, "The Effect of Nodes Density on Real Time Traffic in Mobile Ad Hoc Network," *2019 IEEE 12th International Conference on Global Security, Safety and Sustainability (ICGS3)*, London, United Kingdom, 2019, pp. 209-212.

[50] Riad, A., Elminir, H., Shehata, A. & Ibrahim, T., 2013. SECURITY EVALUATION AND ENCRYPTION EFFICIENCY ANALYSIS OF RC4 STREAM CIPHER FOR CONVERGED NETWORK APPLICATIONS. *Journal of ELECTRICAL ENGINEERING*, 64(3), pp. 196-200.

[51] Rana, B. & Wankhade, S., 2017. Hybrid Cryptographic Algorithm for Enhancing Security of Text. *International Conference On Emanations in Modern Technology and Engineering*, 5(3), pp. 339-443.

[52] Yussra Majid Hameed, Nada Hussein M. Ali, 2019, "Enhanced RC5 Key Schedule Using One-Dimensional Cellular Automata for Audio File Encryption", Iraqi Journal of Science, Volume 60, No.2, pp. 388-401.

[53] Nada Hussein M. Ali Ruaa A. Abdul-Sattar, 2017, "Data integrity enhancement for the encryption of color images based on CRC64 technique using multiple look-up tables", Iraqi Journal of Science, Volume 58, No.3C, pp. 1729-1739.

[54] B. Hammi, R. Khatoun and S. Zeadaly, "IoT technologies for smart cities," *IET networks*, vol. 7, no. 1, pp. 1-13, 2017.