

A NEW SYMMETRICAL ENCRYPTION ALGORITHM WITH HIGH SECURITY AND DATA RATE FOR WLAN

G. RAMESH*

Dr. R. UMARANI**

Abstract:

The hacking is the greatest problem in the wireless local area network (WLAN). Many algorithms like DES, 3DES, AES, CAST, UMARAM and RC6 have been used to prevent the outside attacks to eavesdrop or prevent the data to be transferred to the end-user correctly. The authentication protocols have been used for authentication and key-exchange processes. A new symmetrical encryption algorithm is proposed in this paper to prevent the outside attacks to obtain any information from any data-exchange in Wireless Local Area Network (WLAN). The new symmetrical algorithm avoids the key exchange between users and reduces the time taken for the encryption, decryption, and authentication processes. It operates at a data rate higher than DES, 3DES, AES, UMARAM and RC6 algorithms. It is applied on a text file and an image as an application. The encryption becomes more secure and high data rate than DES, 3DES, AES, CAST, UMARAM and RC6. This new algorithm is designed and implemented by g.ramesh et al. in the year 2010 to prevent the outside attacks to obtain any information from any data-exchange in Wireless Local Area Network.

A comparison has been conducted for those encryption algorithms at different settings for each algorithm such as different sizes of data blocks, different data types, battery power consumption, different key size and finally encryption/decryption speed. Experimental results are given to demonstrate the effectiveness of each algorithm.

Keywords: Plaintext; Encryption; Decryption; S-Box; Key updating; Outside attack; key generation for Proposed Algorithm;

* Research Scholar, Research And Development Centre, Bharathiyar University, Coimbatore, India.

** Associate Professor in Computer Science, Sri Sarada college for women, Salem -16.

1. INTRODUCTION

Wireless Local Area Network (WLAN) is one of the fastest growing technologies. Wireless Local Area Network(WLAN) is found in the office buildings, colleges, universities, and in many other public areas [1]. The security in WLAN is based on cryptography, the science and art of transforming messages to make them secure and immune to attacks by authenticating the sender to receiver within the WLAN.

The cryptography algorithms are divided into two groups: symmetric-encryption algorithms and asymmetric-encryption algorithms. The most common classification of encryption techniques can be shown in Figure 1.

There are a lot of symmetric-encryption algorithms used in WLAN, such as DES [2], TDES [3], AES [4], CAST-256,RC6 [5] and UMARAM[6]. In all these algorithms, both sender and receiver have used the same key for encryption and decryption processes respectively. The attacks on the security of WLAN depend on viewing the function of the computer system in WLAN as providing information (such as company title, the data type can be transferred in WLAN, and the algorithms and authentication protocol used in WLAN). Each

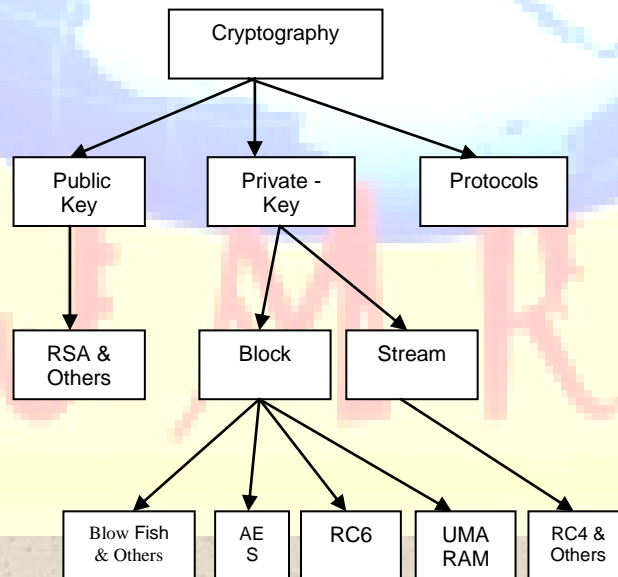


Figure 1: Overview of the field of Cryptography

company sends its title with each message. The outside attacks can use this fixed plaintext, company-title, and encrypted text of that title to obtain the key used in WLAN. The outside

attack can also appear as a fox because he can lie to use a computer on the WLAN to send an important message to someone because there are some troubles in his device while his device is still open to take a copy from the encrypted message. The plaintext and encrypted text are known. He can obtain the key used for encryption and decryption processes easily. The authentication protocols have been used for authentication and key-exchange processes, such as EAP-TLS [9], EAP-TTLS [9], and PEAP [10]. The attacker can be authorized-user and he will be accepted to access the network after the success of authentication and key exchange processes. He will act as an evil to analysis the data-exchange to eavesdrop or act as man-in-the-middle. The proposed algorithm will avoid key-exchange, the time taken for authentication process, and it will avoid the foxes.

This paper examines a method for evaluating performance of selected symmetric encryption of various algorithms. Encryption algorithms consume a significant amount of computing resources such as CPU time, memory, and battery power. Battery power is subjected to the problem of energy consumption due to encryption algorithms. Battery technology is increasing at a slower rate than other technologies. This causes a "battery gap" [17, 18]. We need a way to make decisions about energy consumption and security to reduce the consumption of battery powered devices.

This study evaluates seven different encryption algorithms namely; AES, DES, 3DES, RC6, Blowfish, UMARAM and RC2. The performance measure of encryption schemes will be conducted in terms of energy, changing data types - such as text or document, Audio data and video data power consumption, changing packet size and changing key size for the above and proposed cryptographic algorithms.

This paper is organized as follows. Section 2 gives a short review of the symmetrical-encryption algorithms and authentication protocols. Section 3 presents the proposed algorithm. Section 4 shows the results. Section 5 presents Experimental design of Metrics of our proposed algorithm. Section 6 presents Experimental result. Conclusions are presented in section 5.

2. REVIEW ON THE SYMMETRICAL-ENCRYPTION ALGORITHMS AND AUTHENTICATION PROTOCOLS:

There are a lot of the symmetrical-encryption algorithms in WLAN. The Data Encryption Standard [2], known as Data Encryption Algorithm (DEA) by the ANSI [11] and the DEA-1 by the ISO [12] remained a worldwide standard for a long time and was replaced by the new Advanced Encryption Standard (AES). However, it is expected that DES will remain in the public domain for a number of years [12]. It provides a basis for comparison for new algorithms and it is also used in. DES is a block cipher symmetric algorithm; the same data processing and key are used for both encryption and decryption. The basic building block (a substitution followed by a permutation) is called a round and is repeated 16 times [2]. For each DES round, a sub-key is derived from the original key using an algorithm called key schedule. Key schedule for encryption and decryption is the same except for the minor difference in the order (reverse) of the sub-keys for decryption. In the encryption process, DES encrypts the data in 64-bit blocks using a 64-bit key (although its effective key length is in reality only 56-bit).

Triple DES, TDES, is a block cipher formed from the DES cipher by using it three times. When it was found that a 56-bit key of DES is not enough to guard against brute force attacks, TDES was chosen as a simple way to enlarge the key space without a need to switch to a new algorithm. The use of three steps is essential to prevent the man-in-the-middle attacks that are effective against double DES encryption. The simplest variant of TDES encryption operates as follows: $DES(k_3; DES^{-1}(k_2; DES(k_1; M)))$, where M is the message block to be encrypted, k_1 , k_2 , and k_3 are DES keys, and DES and DES^{-1} refer to the encryption and decryption modes respectively. While the TDES decryption operates as follows: $DES^{-1}(k_1; DES(k_2; DES^{-1}(k_3; C)))$, where C is the cipher text block.

The Advanced Encryption Standard (AES) algorithm is a symmetric block. AES algorithm can encrypt and decrypt the plaintext and cipher text of 128-bits respectively by using cryptographic keys of 128-bits (AES-128), 192-bits (AES-192), or 256-bits (AES-256). Number of rounds in the encryption or decryption processes depends on the key size. CAST-256 belongs to the class of encryption algorithms known as Feistel ciphers; overall operation is thus similar to the Data Encryption Standard (DES). The algorithm was created by Carlisle Adams and Stafford Tavares. It is a symmetric block. RC6 is more accurately specified as RC6-w/r/b where the

word size is w bits, encryption consists of a nonnegative number of rounds r , and b denotes the length of the encryption key in bytes. Since the AES submission is targeted at $w = 32$ and $r = 20$, RC6 shall be used as shorthand to refer to such versions. When any other value of w or r is intended in the text, the parameter values will be specified as RC6- w/r . Of particular relevance to the AES effort will be the versions of RC6 with 16-, 24-, and 32-byte keys.

The UMARAM is a Symmetrical encryption algorithm. The key generation generates 16-keys during 16-rounds. One key of them is used in one round of the encryption or decryption process. The new algorithm uses a key size of 512-bits to encrypt a plaintext of 512-bits during the 16-rounds. In this Algorithm, a series of transformations have been used depending on S-BOX, different shift processes, XOR-Gate, and AND-Gate. The S-Box is used to map the input code to another code at the output. It is a matrix of $16 \times 16 \times 16$. The S-Box consists of 16-slides, and each slide having 2-D of 16×16 . The numbers from 0 to 255 are arranged in random positions in each slide.

The Authentication Protocols are used for Authentication and key-exchange processes to avoid the outside attacks to access the network. The researchers have researched on the best authentication protocol to authenticate the overall devices in the network and prevent the attacks to effect on the network or eavesdropping on the interchangeable data. The Extensible Authentication Protocol (EAP) [13] is an authentication framework which supports multiple authentication methods. EAP typically runs directly over data link layers such as Point-to-Point Protocol (PPP) [14] or IEEE 802.11 [15], without requiring IP [1]. EAP has been implemented with hosts and routers that connect via switched circuits or dial-up lines using PPP. It has also been implemented with switches and access points using IEEE 802.11. EAP-TLS EAP-TTLS are EAP methods used for WLAN authentication and key derivation. EAP-TTLS provides additional functionality beyond what is available in EAP-TLS. There are a lot of Authentication protocols used for WLAN authentication and key derivation but, the proposed algorithm will avoid the key derivation and reduce the delay time for authentication process, as the following sections.

3. PROPOSED SYMMETRICAL ALGORITHM:

A block encryption algorithm is proposed in this approach. In this Algorithm, a series of transformations have been used depending on S-BOX, XOR Gate, and AND Gate. The proposed algorithm encrypts a plaintext of size 64-bits by a key size of 64-bits. It uses eight rounds for encryption or decryption process. It overcomes some drawbacks of the other algorithms. It is more efficient and useable for the Wireless Local Area Network because it avoids the using of the same key with other packets within a message. The algorithm is simple and helpful in avoiding the hackers. S-BOX generation is the backbone of this algorithm. It has eight columns and 256 rows; each element consists of 8-bits, see Appendix A for the contents of S-boxes. It replaces the input by another code to the output. The order of the columns is changed in each round as follows:

Round 1: C1C2C3C4C5C6C7C8

Round 2: C2C3C4C1C8C5C6C7

Round 3: C3C4C1C2C7C8C5C6

Round 4: C4C1C2C3C6C7C8C5

Round 5: C5C8C7C6C3C2C1C4

Round 6: C6C5C8C7C2C1C4C3

Round 7: C7C6C5C8C1C4C3C2

Round 8: C8C7C6C5C4C3C2C1

Figure (2) combines between keys generation and Data encryption. There are two external inputs for keys generation, R_{ni} and R_v , where i is the round number, $i=1, \dots, 8$. R_v has two hexadecimal values, (00 00 00 00 00 00 00 00) and (FF FF FF FF FF FF FF FF). R_{ni} has two hexadecimal values, (00 00 00 00 00 00 00 00) and the initial key value, 64-bits, used at the first time. The initial key, 64-bits, can be the same for all rounds or each round can have different initial key as the designer like.

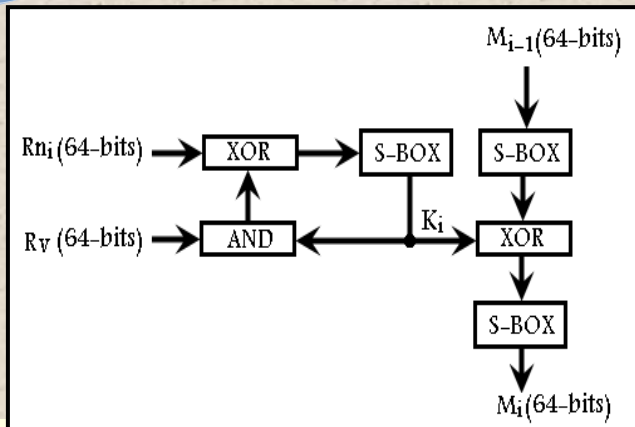


Figure (2): Proposed Algorithm for Encryption or Decryption

Round-Key generation, at the first time, begins by using the first value of R_v , (00 00 00 00 00 00 00 00), to avoid any noise from the feedback of the S-BOX to the initial key. R_{ni} equals, at the first time, to initial value of round-key. Then, the initial value of the key, 64-bits is divided into eight parts, 8-bits each. Each part will travel to a row, under the same column, having a number equals to its number plus one, it will find a code, 8-bits, that will be used instead of this part. For example, part 3=A2, it will take the code in the column number 3, according to the columns-ordering of the S-BOX in each round, and the row number $162+1=163$ and column number 3, where A2 in a hexadecimal format equals to 162 as a decimal value. The eight parts will be replaced by another eight parts, they will be used as a round-key to encrypt the message in each round, and also will feedback to update the round-key to another key by changing R_v to its second value, (FF FF FF FF FF FF FF FF), and also R_{ni} to the other value, (00 00 00 00 00 00 00 00). So the algorithm will update its round-key by itself, and each round will choose its key randomly from $2^{64} = 18,446,744,073,709,551,616$ available keys. Thus, in each encryption process, a different key will be used for each round; it gives the impossibility to the hackers to decrypt the cipher text. The R_v initial value, (00 00 00 00 00 00 00 00), is used to make synchronization between the transmitter and receiver when there are troubles appeared in the decryption process, the receiver must send a message to the transmitter to request the reset of R_v value, in this case $R_v = (00 00 00 00 00 00 00 00)$, and the R_{ni} must equal to the initial key value. Otherwise, $R_v = (FF FF FF FF FF FF FF FF)$ and $R_{ni} = (00 00 00 00 00 00 00 00)$.

In the data encryption, as round-key generation, the message block, 64-bits, is divided into eight parts to apply them to the eight columns of the S-BOX. The order of column depends on the round number. The output of the S-BOX will be XORed with round-key. The output of the XOR gate will be divided into to eight parts to apply to the S-BOX. The encrypted block will be the input of the next round, see figure (2), The Key generation of each round does not depend on the other round-key generation. The data decryption process is the same as the data encryption process but, the order of the round-key, K_j , used in the encryption process will be reversed to be used in the decryption process, and cipher text becomes instead of the plaintext to obtain the decrypted block as the same as the plaintext. The key will be updated by itself and the next packet will use different key. Each round will use different key because the order of columns of the S-BOX is interchanged. If there are NAK from the receiver, the sender will encrypt the packet by the initial key, default case, by applying $R_{ni} = \text{Initial key}$, and $R_{v=} = (00\ 00\ 00\ 00\ 00\ 00\ 00\ 00)$ to reset the system to the default case. If the outsider attack prevents any packet or message to reach the receiver, the next packet or message can not be decrypted correctly because at this situation the key used for encryption is not the same as that used for decryption and these will be no synchronization between the sender and the receiver. The receiver will know that there is something wrong in the transmitted message because of virus, outside attacks, or environment noise to reach correctly. The receiver will send NAK to the sender. The NAK is a message of all 0-bits and the number of the damaged packet. The NAK length is 64-bits as the normal message. The NAK will be encrypted by the last updated-key, as the normal message will be encrypted, to avoid the traffic analysis from the outsider attacks.

This initial key is used only in three cases, the connection in the first time, NAK, and authentication process. In authentication process, the sender and the receiver will interchange a secret message encrypted by last updated key. If this message encrypted again, the encrypted message will have a different contents than the first one. The outside attack can not find out the key even if he knows the title of the company because the encrypted title will take other form and the key- generation of each round does not depend on each others. The designer can use different initial keys for each round to make the system more secure.

4. RESULTS:

The proposed algorithm is applied on a text by using:

A. Software

- Microsoft Visual C++ Program.

B. Hardware

- Intel(R) Pentium(R) 4 CPU 2.8GHz
- 1GB of RAM

The plain text, the encrypted text, the decrypted text are shown in figure (3a, b, and c). When the same text is encrypted again, a different encrypted text is obtained, see figure (3d), it means that, the key is updated in each round.

This paper has proposed a block encryption algorithm using S-Box and XOR gate. The system becomes more secure because of key-updating with each packet. It is simple and the delay time will be reduced than DES, 3DES, AES, and RC6 algorithms because of no multiple functions used. The outsider attacks can not know the key even if they have the plaintext and the cipher text. The algorithm will help the authentication protocols to reduce the delay taken by them, and gives the channel the data security wanted. The programs ensure the key updated without any problem on the decryption of the text or the image, and show that the algorithm reduce the time used in the encryption or decryption process. It is efficient and useable for the security in the WLAN systems.

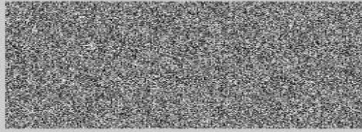
Figure (3 a): The Plain Text

駒 馱 鮑 闕 除 鑛 營 銷 公 司 形 鑛 業 模 倉 蜂 蝦 均 櫛 竈 輓
 き 霽 寔 義 17 礎 邇 峯 蔭 境 鈔 象 扇 嘔 噉 熾
 矣 語 襪 猥 香 嶂 嶺 嶺 之 邑 誌 卷 詠 軀 盜 鈔 鈔
 君 寧 擲 遍 櫛 櫛 滿 人 忌 輸 吉 灼 檄 欬 卍 卍
 厔 搥 繪 調 達 裂 不 穢 糶 輶 覬 規
 罽 嬰 仰 愜 裱 襪 襪 糶 嬰 嬰 嬰 莠 莠
 办 擦 須 糶 5 吳 粘 婦 鴉 鴉 鴉 鴉 鴉 鴉 鴉 鴉 鴉 鴉 鴉 鴉
 鴉 鴉 鴉 鴉 鴉 鴉 鴉 鴉 鴉 鴉 鴉 鴉 鴉 鴉 鴉 鴉

Figure (3 .b): The Cipher Text



a. The Plain-Image



b. The Encrypted-image



c. The decrypted-Image

Figure (4): Black and white Image



a. The Plain Image



b. The Decrypted Image

Figure (5): The Color Image

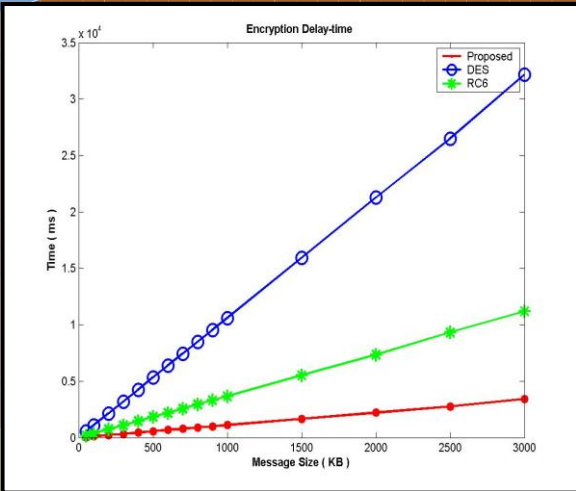


Figure (6): The Encryption delay time for DES, RC6 and the proposed algorithms

The encryption and decryption processes are applied between two wireless computers in WLAN, their specifications are:

- PC1: Intel(R) Pentium(R) 4 CPU 2.8GHz, and 1GB of RAM.
- PC2: Intel(R) Pentium(R) 4 CPU 2GHz, and 256 MB of RAM.
- A 54M Wireless Access Point of TP-Link (TL-WA501G).
- Two 54M Wireless USB Adapter of TP-Link (TL-WN322G).

The encryption and the decryption of the text, black and white image, and the color image are done successfully, see figure (7).

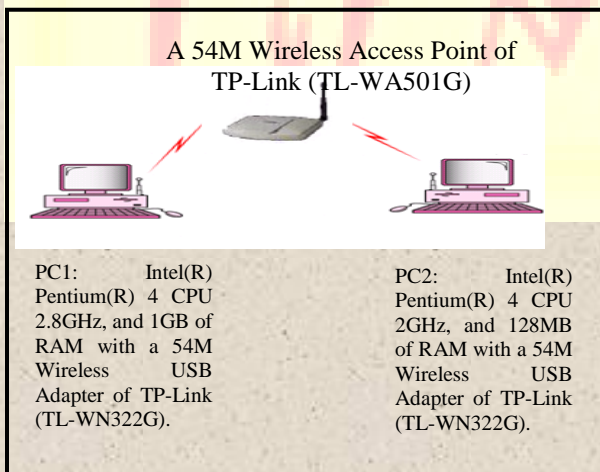


Figure (7): Wireless LAN (infrastructure mode)

As an improvement in The proposed algorithm to be extra more secure. . The size of the plain text can be varied and the network administrator generates the S-Box by himself. First, the plain text size must be specified to satisfy the equation (1a).

$$\text{The Plain text size} = 8 \times n \text{ -----(1a)}$$

Where n is the number of rounds used in the encryption or decryption process and is an integer, even, non-zero, and positive number so, $n=2,4,6,\dots$, see Figure (8). Second, the Administrator will generate his Network S-Box according to the flowchart of Figure (8), where r and c are the row and column number respectively of the S-Box, and they start from 1 to n. The S-Box Generation starts by generating a random number between 0 and 255 for each element in the S-Box matrix. No number-repetition can be found in each column. .

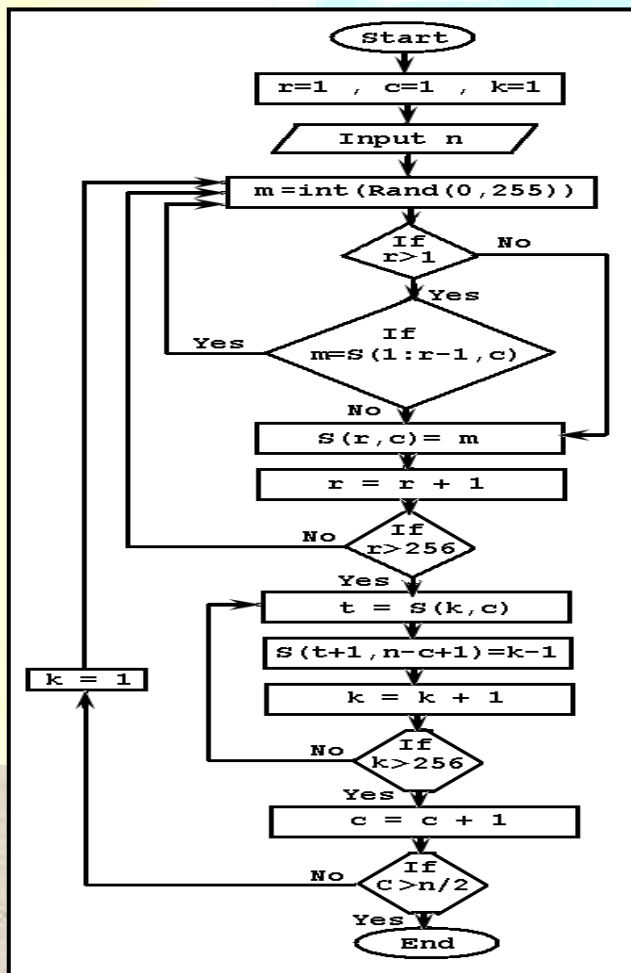


Figure (8): S-Box Generation Flow Chart

The generated S-Box has a size of $256*n$. The order of the columns in the S-Box is changed in each round according to the diagram of the figure (9). For example, if $n=8$, the Columns orders of the round 1 to round 4 are:

Round 1: C1C2C3C4C5C6C7C8

Round 2: C2C3C4C1C8C5C6C7

Round 3: C3C4C1C2C7C8C5C6

Round 4: C4C1C2C3C6C7C8C5

After the round $(n/2)$, the first part and the second part are interchangeable and are mirrored, and also the shift direction is changed, see figure (9). Thus, the Columns orders of the round 5 to round 8 are:

Round 5: C5C8C7C6C3C2C1C4

Round 6: C6C5C8C7C2C1C4C3

Round 7: C7C6C5C8C1C4C3C2

Round 8: C8C7C6C5C4C3C2C1

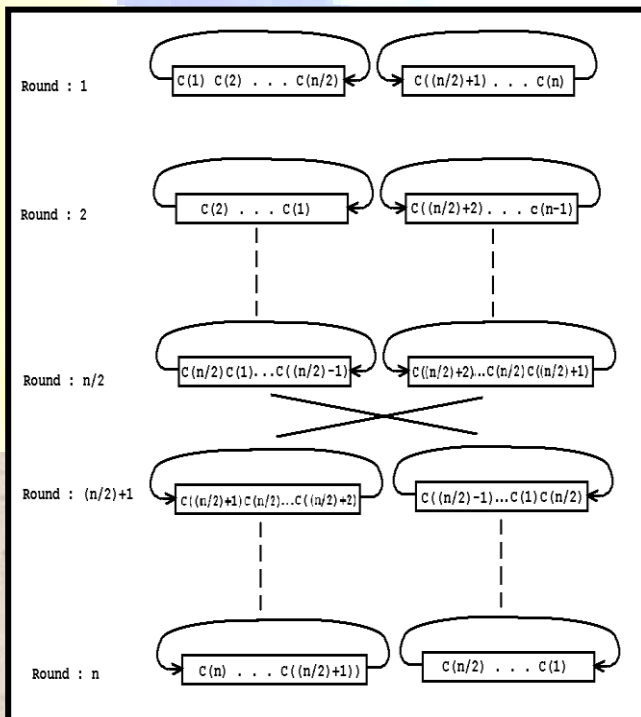


Figure (9): The relationship between Round number and the Column orders of S-Box

Thus, the system will be more secure because of the following reasons.

1. The S-Box generation can be generated from the administrator himself.
2. The delay time taken for the encryption and the decryption processes by the proposed algorithm is less than the time taken **DES, 3DES, AES, UMARAM and RC6** algorithms.
3. Higher data rate than DES, 3DES, AES, UMARAM and RC6 algorithms.
4. The initial key can be chosen from any row in the S-box, and the authentication protocol will interchange the row number of the unknown S-box to be used as a key instead of key interchanging. It will keep the key in the system and prevent it to fly between the network devices.
5. Each round can use special initial key and they are independent.
6. The NAK becomes unknown to the outsider attacks.
7. The outside attacks can not obtain the key or any information about the algorithm even if he had the plaintext, the company title, S-Box, and the encrypted message because they will loss the synchronization or the initial key of each round where they are independent.

In addition, the proposed algorithm has the following advantages:

- Simple.
 - The updating of the round-key with each packet.
 - The encryption and decryption processes are the same.
 - Any change in the transmitted message will be known to the sender and the receiver, so it will prevent the attacks such as, man-in-the middle attacks to analysis the traffic or decrypt the encrypted message, and the foxes to have the key.
-
- Our proposed algorithm can meet the growth of the technology.

5. EXPERIMENTAL DESIGN FOR METRIC OF PROPOSED SYSTEM:

For our experiment, we use a laptop IV 2.4 GHz CPU, in which performance data is collected. In the experiments, the laptop encrypts a different file size ranges from 321 K byte to 7.139Mega Byte139MegaBytes for text data, from 33 Kbytes to 8262 Kbytes for audio data, and from 4006 Kbytes to 5073 Kbytes for video files.

Several performance metrics are collected: 1) Encryption time; 2) CPU process time; and 3) CPU clock cycles and battery power,4)Throughput,5)Different data types,6)Different size of data block.

5.1 Measurement of Throughput:

The encryption time is considered the time that an encryption algorithm takes to produce a cipher text from a plaintext. Encryption time is used to calculate the throughput of an encryption scheme. It indicates the speed of encryption. The throughput of the encryption scheme is calculated as the total plaintext in bytes encrypted divided by the encryption time [19].

Throughput=Total plaintext encrypted in bytes / Encryption time

The CPU process time is the time that a CPU is committed only to the particular process of calculations. It reflects the load of the CPU. The more CPU time is used in the encryption process, the higher is the load of the CPU. The CPU clock cycles are a metric, reflecting the energy consumption of the CPU while operating on encryption operations. Each cycle of CPU will consume a small amount of energy.

The encryption time is considered the time that an encryption algorithm takes to produce a cipher text from a plaintext. Encryption time is used to calculate the throughput of an encryption scheme. It indicates the speed of encryption. The throughput of the encryption scheme is calculated as in equation (1b).

$$\text{Throughput of Encryption} = \frac{T_p (\text{Bytes})}{E_t (\text{Second})} \text{----- (1b)}$$

Where T_p – Total Plain Text (bytes), and E_t - Encryption Time (Second)

The CPU process time is the time that a CPU is committed only to the particular process of calculations. It reflects the load of the CPU. The CPU clock cycles are a metric, reflecting the energy consumption of the CPU while operating on encryption operations. Each cycle of CPU will consume a small amount of energy.

5.2 Measurement of Energy Consumption

The study of the energy consumption of the encryption schemes in wireless devices is essential in design of energy efficient security protocols customized to the wireless environment. A key limitation in wireless devices is the battery capacity, while memory and processor technologies double with the introduction of every new semiconductor generation (roughly every 18 months) [7]; battery technology is increasing at the much slower rate of 5%-10% per year. This is causing a gap to form between the power required and the battery available [7].

Energy consumption for encryption and decryption can be measured in many ways. These methods as follows:

The First method used to measure energy consumption is to assume that an average amount of energy is consumed by normal operations and to test the extra energy consumed by an encryption algorithms. This method simply monitors the level of the percentage of remaining battery that can computed by equations (2), (3)

The battery life consumed in percentage for one run = $\frac{\text{Change in Battery Life}}{\text{the number of runs}}$ -----(2)

Average battery Consumed per iteration = $\sum_{i=1}^N \left(\frac{\text{Battery Consumed Per iteration}}{N} \right)$ -----(3)

The second method of security primitives can also be measured by counting the amount of computing cycles which are used in computations related to cryptographic operations. For computation of the energy cost of encryption, we use the same techniques as described in using the following equations.

$$Bcost_encryption (ampere-cycle)= \tau * I ----- (4)$$

$$Tenergy_cost(ampere-seconds) = \frac{Bcost_encryption(ampere-cycle)}{F(cycles/sec)} -----(5)$$

$$Ecost(Joule) = Tenergy_cost (ampere-seconds) * V---(6)$$

Where Bcost-encryption: a basic cost of encryption(ampere-cycle): the total number of clock cycles.

I : the average current drawn by each CPU clock cycle

Tenergy_cost : the total energy cost(ampere-seconds), F : Clcok frequency(cycles/sec)

Ecost(joule) : the energy cost(consumed)

By using the cycles, the operating voltage of the CPU, and the average current drawn for each cycle, we can calculate the energy consumption of cryptographic functions. For example, on average, each cycle consumes approximately 270 mA on an Intel IV 2.4 MHz processor or 180 mA on Intel Strong ARM . For a sample calculation, with a 700 MHz CPU operating at 1.35 Volt, an encryption with 20,000 cycles would consume about 5.71×10^{-3} mA-second or 7.7 μ Joule. So, the amount of energy consumed by program P to achieve its goal (encryption or decryption) is given by

$$E= VCC \times I \times N \times \tau -----(7)$$

Where N – The number of clock cycles and VCC – The supply voltage of the system

I – the average current in amperes drawn from the power source for T seconds.

Since for a given hardware, both VCC and τ are fixed, $E \propto I \times N$. However, at the application level, it is more meaningful to talk about T than N, and therefore, we express energy as $E \propto I \times T$. Since for a given hardware Vcc are fixed [22]. The Second and third methods were used in this work. A comparison is conducted between the results of selected different encryption algorithms using different setting such as different data types, different packet size, different key size.

The following tasks that will be performed are shown as follows:

- A comparison is conducted between the results of the selected different encryption and decryption schemes in terms of the encryption time at two different encoding bases namely; hexadecimal base encoding and in base 64 encoding.
- A study is performed on the effect of changing packet size at power consumption during throughput for each selected cryptography algorithm.
- A study is performed on the effect of changing data types - such as text or document, audio file, and video file for each cryptography selected algorithm on power consumption.
- A study is performed on the effect of changing key size for cryptography selected algorithm on power consumption.

6. DIFFERENT METRICS OF PROPOSED ALGORITHM:

6.1 Differentiate Output Results of Encryption (Base 64, Hexadecimal)

Experimental results are given in Figures 10 and 11 for the selected seven encryption algorithms at different encoding method. Figure 9 shows the results at base 64 encoding while Figure 10 gives the results of hexadecimal base encoding. We can notice that there is no significant difference at both encoding method. The same files are encrypted by two methods; we can recognize that the two curves almost give the same results.

Time consumption of encryption algorithm (base 64 encoding)

6.2 Effect of Changing Packet Size for Cryptographic Algorithms on Power Consumption

6.2.1 Encryption of Different Packet Size

Encryption time is used to calculate the throughput of an encryption scheme. The throughput of the encryption scheme is calculated by dividing the total plaintext in Megabytes encrypted on

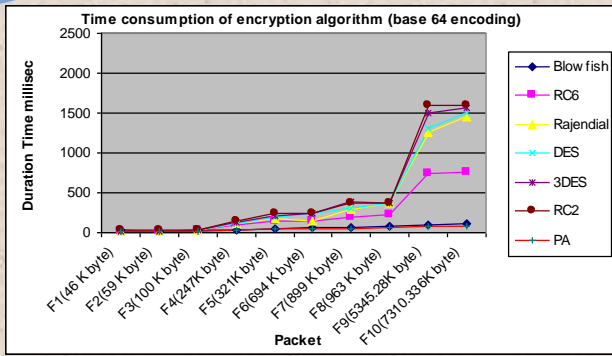


Figure 9: Time consumption of encryption algorithm

(base 64 encoding)

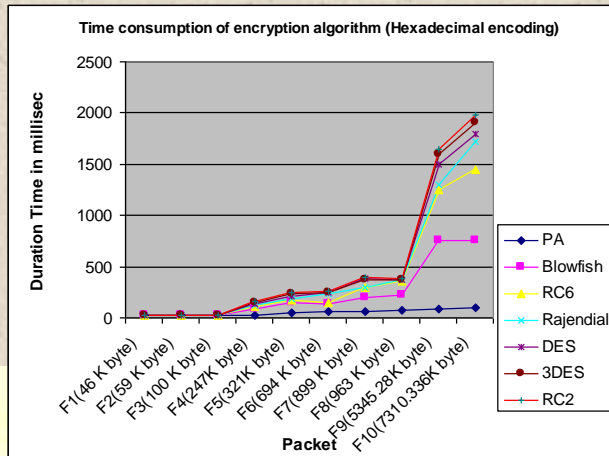
the total encryption time for each algorithm in. As the throughput value is increased, the power consumption of this encryption technique is decreased.

Experimental results for this concern point are shown Figure 11 at encryption stage. The results show the advantage of Proposed algorithm over other algorithms in terms of the processing time. Another point can be noticed here; that RC6 requires less time than all algorithms except Proposed Algorithm. A third point can be noticed here; that AES has an advantage over other 3DES, DES and RC2 in terms of time consumption and throughput. A fourth point can be noticed here; that 3DES has low performance in terms of power consumption and throughput when compared with DES. It always requires more time than DES because of its triple phase encryption characteristics. Finally, it is found that RC2 has low performance and low throughput when compared with other six algorithms in spite of the small key size used.

6.2.2 Decryption of Different Packet Size

Experimental results for this compassion point are shown Figure 12 decryption stage. We can find in decryption that Proposed Algorithm is the better than other algorithms in throughput and power consumption. The second point should be noticed here that RC6 requires less time than all algorithms except Proposed Algorithm. A third point that can be noticed that AES has an advantage over other 3DES, DES, RC2. The fourth point that can be considered is that RC2 still

has low performance of these algorithm. Finally, Triple DES (3DES) still requires more time



than DES.

Figure 10: Time consumption of encryption algorithm (Hexadecimal encoding)

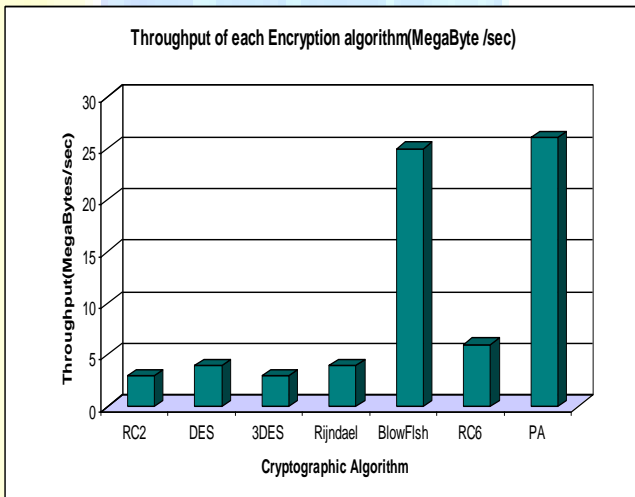


Figure 11: Throughput of each encryption algorithm (Megabyte/Sec)

6.3 The Effect of Changing File Type (Audio Files) for Cryptography Algorithm on Power Consumption

6.3.1 Encryption of Different Audio Files (Different Sizes) Encryption Throughput

In the previous section, the comparison between encryption algorithms has been conducted at text and document data files. Now we will make a comparison between other types of data (Audio file) to check which one can perform better in this case. Experimental results for audio data type are shown Figure 13 at encryption.

CPU Work Load

In Figure 14, we show the performance of cryptographic algorithms in terms of sharing the CPU load. With a different audio block size Results show the superiority of Proposed algorithm over other algorithms in terms of the processing time (CPU work load) and throughput. Another point can

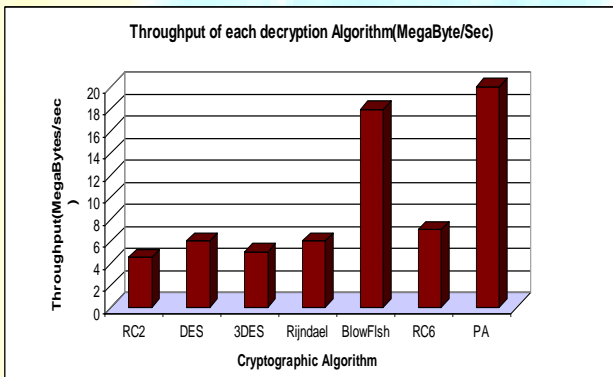


Figure12: Throughput of each decryption algorithm (Megabyte/Sec)

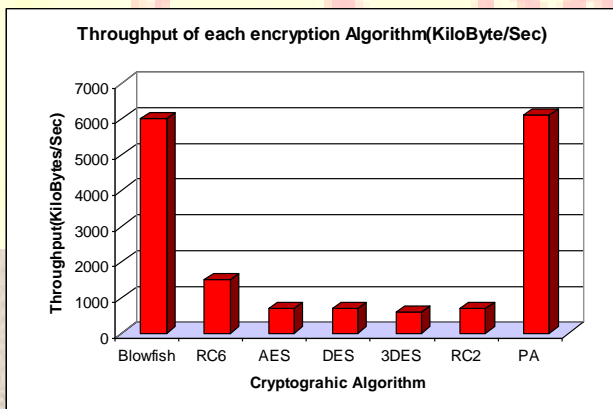


Figure 13: Throughput of each encryption algorithm (Kilo-bytes/Second)

be noticed here; that RC6 requires less time than all algorithms except Proposed Algorithm. A third point can be noticed here; that AES has an advantage over other 3DES, DES and RC2 in terms of time consumption and throughput especially in small size file.

A fourth point can be noticed here; that 3DES has low performance in terms of power consumption and throughput when compared with DES. It always requires more time than DES. Finally, it is found that RC2 has low performance and low throughput when compared with other six algorithms in spite of the small key size used.

Decryption of Different Audio files (Different Sizes)

Decryption Throughput Experimental results for this comparison point are shown Figure 15.

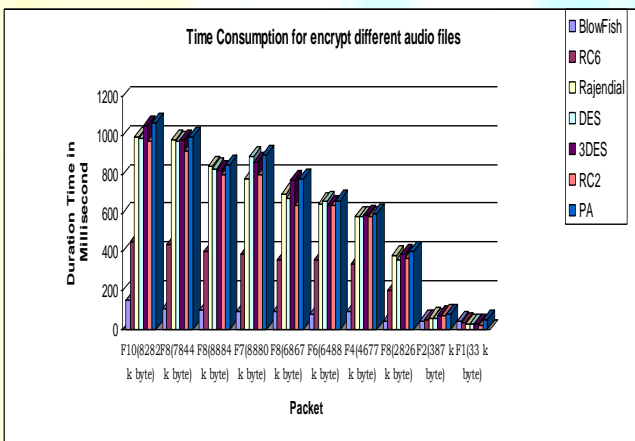


Figure 14: Time consumption for encrypt different audio files

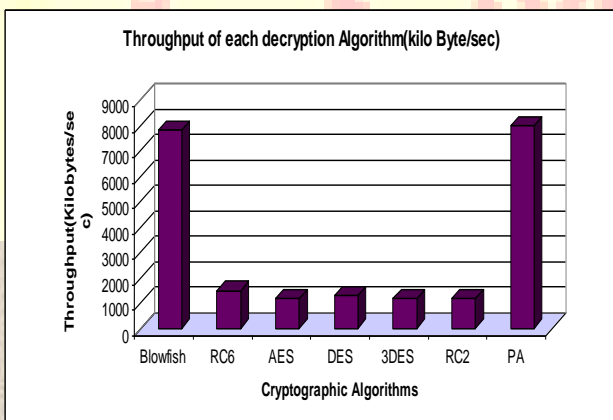


Figure 15: Throughput of each Decryption algorithm (Kilobytes / Second)

CPU Work Load

Experimental results for this compassion point are shown Figure 16. From the results we found the result as the same as in encryption process for audio files.

6.4 The Effect of Changing File Type (Video Files) for Cryptography Algorithm on Power Consumption

6.4.1 Encryption of different video files (different sizes)

Encryption Throughput

Now we will make a comparison between other types of data (video files) to check which one can perform better in this case. Experimental results for video data type are shown Figure 17 at encryption.

CPU Work Load

In Figure 18, we show the performance of cryptography algorithms in terms of sharing the CPU load. With a different audio block size.

The results show the superiority of Proposed algorithm over other algorithms in terms of the processing time and throughput as the same as in Audio files. Another point can be noticed here; that RC6 still requires less time has throughput greater than all algorithms except Proposed Algorithm. A third point can be noticed here; that 3DES has low performance in terms of power consumption and throughput when compared with DES. It always requires more time than DES. Finally, it is found that RC2 has low performance and low throughput when compared with other six algorithms.

6.4.2 Decryption of Different Video Files (Different Sizes)

Decryption Throughput

Experimental results for this comparison point are shown Figure 19.

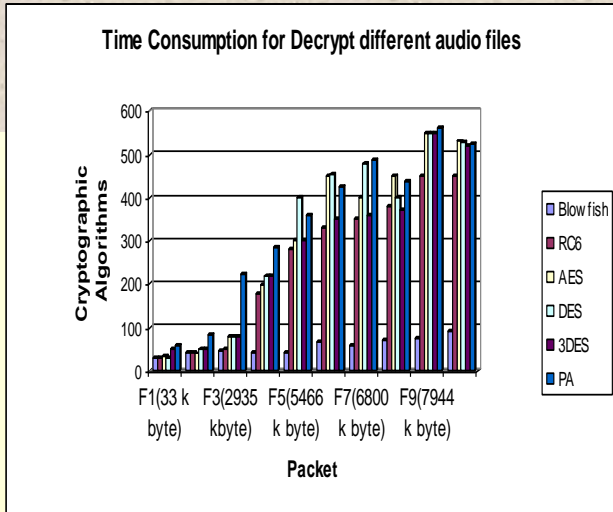


Figure 16: Time consumption for decrypt different audio files

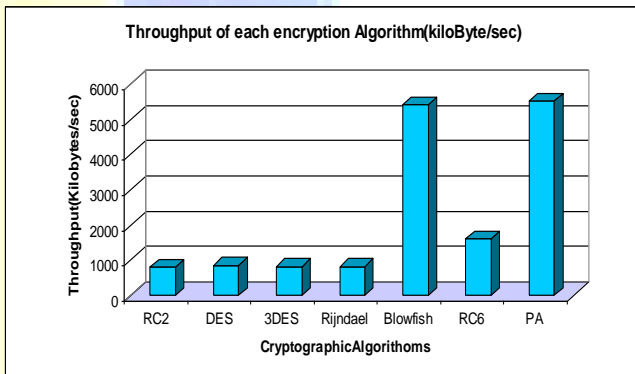


Figure 17: Throughput of each encryption algorithm (Kilobytes/sec)

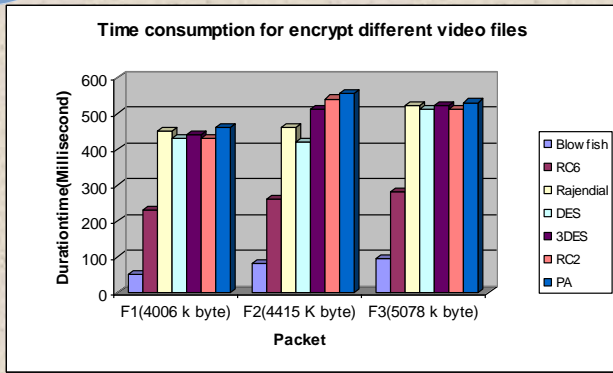


Figure 18: Time consumption for encrypt different video files

CPU Work Load

Experimental results for this compression point are shown Figure 20. From the results we found the result as the same as in encryption process for video and audio files.

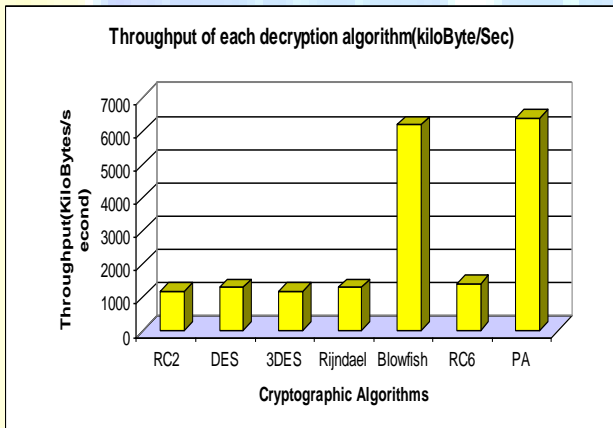


Figure 19: Throughput of each decryption algorithm
(Kilobytes/Second)

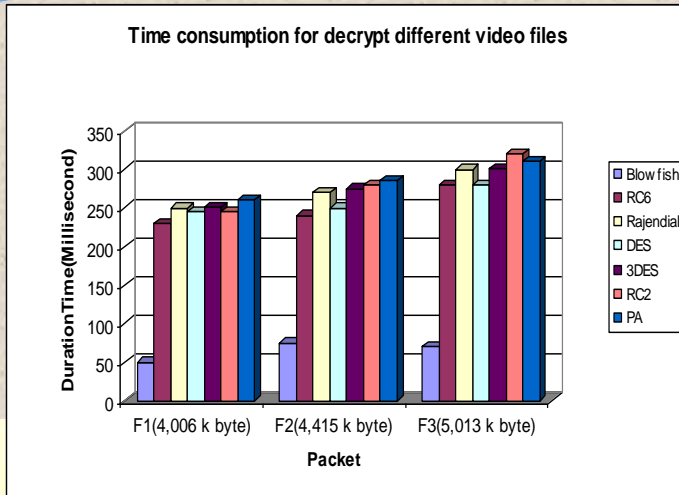


Figure 20: Time consumption for decrypt different video files

6.5 The Effect of Changing Key Size of AES, And RC6 on Power Consumption

The last performance comparison point is changing different key sizes for AES and RC6 algorithm. In case of AES, we consider the three different key sizes possible i.e., 128-bit, 192-bit and 256-bit keys. The Experimental result are shown in Figures 21 and 22.

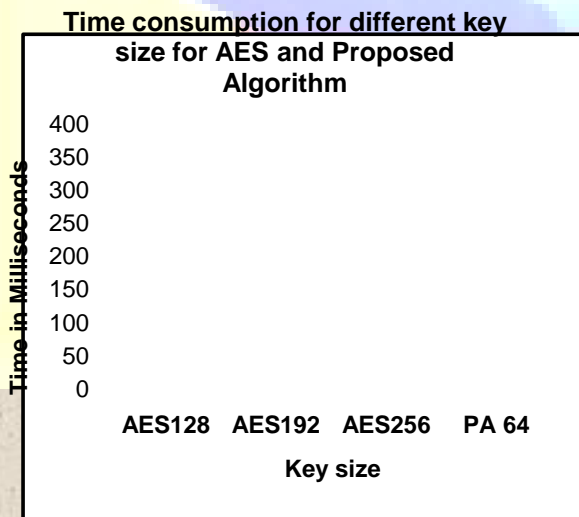


Figure 21: Time consumption for different key size for AES and PA

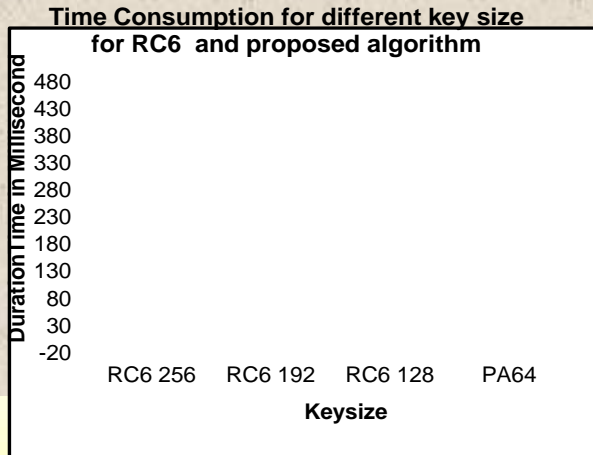


Figure 22: Time consumption for different key size for RC6 and PA

In case of AES it can be seen that higher key size leads to clear change in the battery and time consumption. It can be seen that going from 128-bit key to 192-bit causes increase in power and time consumption about 8% and to 256-bit key causes an increase of 16% [12]. Also in case of RC6, we consider the three different key sizes possible i.e., 128-bit, 192-bit and 256-bit keys. The result is close to the one shown in the following figure: In case of RC6 it can be seen that higher key size leads to clear change in the battery and time consumption.

7. CONCLUSION:

This paper has proposed a block encryption algorithm using S-Box and XOR gate. The system becomes more secure because of key-updating with each packet. It is simple and the delay time will be reduced than DES, 3DES, AES, and RC6 algorithms because of no multiple functions used. The outsider attacks can not know the key even if they have the plaintext and the cipher text. The algorithm will help the authentication protocols to reduce the delay taken by them, and gives the channel the data security wanted. The programs ensure the key updated without any problem on the decryption of the text or the image, and show that the algorithm reduce the time used in the encryption or decryption process. It is efficient and useable for the security in the WLAN systems.

The selected algorithms are AES, DES, 3DES, RC6, Blowfish, RC2 and Proposed Algorithm were tested. Several points can be concluded from the Experimental results. Firstly; there is no significant difference when the results are displayed either in hexadecimal base encoding or in base 64 encoding. Secondly; in the case of changing packet size, it was concluded that proposed Algorithm has better performance than other common encryption algorithms used, followed by RC6. Thirdly; we find that 3DES still has low performance compared to algorithm DES. Fourthly; RC2, has disadvantage over all other algorithms in terms of time consumption. Fifthly; we find AES has better performance than RC2, DES, and 3DES. In the case of audio and video files we found the result as the same as in text and document. Finally in the case of changing key size - it can be seen that higher key size leads to clear change in the battery and time consumption.

REFERENCES:

- William Stallings “ Network Security Essentials (Applications and Standards)”, Pearson Education, 2004.
- National Bureau of Standards, “ Data Encryption Standard,” FIPS Publication 46, 1977.
- Jose J. Amador, Robert W.Green, " Symmetric-Key Block Ciphers for Image and Text Cryptography", International Journal of Imaging System Technology,2005.
- Daemen, J., and Rijmen, V. "Rijndael: The Advanced Encryption Standard." Dr. Dobb's Journal, March 2001.
- Adams,C. “ Constructing Symmetric Ciphers Using the CAST Design.” Design, Codes, and Cryptography, 1997.
- Ramesh G, Umarani. R, ” Data Security In Local Area Network Based On Fast Encryption Algorithm”,International Journal of Computing Communication and Information System(JCCIS) Journal Page 85-90. 2010.
- S. Contini, R.L. Rivest, M.J.B. Robshaw and Y.L. Yin. “The Security of the RC6 Block Cipher. Version 1.0 “. August 20, 1998.

- Simon, D., Aboba, B., and R. Hurst, "The EAP-TLS Authentication Protocol", RFC 5216, March 2008.
- P. Funk and S. Blake-Wilson, "EAP Tunneled TLS Authentication Protocol Version 1 (EAP-TTLSv1)", The Internet Society, Mar. 2006.
- Palekar, A., Simon, D., Zorn, G., Salowey, J., Zhou, H., and S. Josefsson, "Protected EAP Protocol (PEAP) Version 2", work in progress, October 2004.
- ANSI3.106, "American National Standard for Information Systems—Data Encryption Algorithm—Modes of Operation," American National Standards Institute, 1983.
- Bruce Schneider, John Wiley & Sons, Inc., "Applied Cryptography, Second Edition," New York, NY, 1996.
- Aboba, B., Blunk, L., Vollbrecht, J., Carlson, J., and H. Levkowitz, "Extensible Authentication Protocol (EAP)", RFC 3748, June 2004.
- Simpson, W., "The Point-to-Point Protocol (PPP)", STD 51, RFC 1661, July 1994.
- Institute of Electrical and Electronics Engineers, "Local and Metropolitan Area Networks: Overview and Architecture", IEEE Standard 802, 1990.
- Aamer Nadeem, Dr M. Younus Javed, "A Performance Comparison of Data Encryption Algorithms", IEEE International Conference on Networking, 2009.
- R. Chandramouli, "Battery power-aware encryption," ACM Transactions on Information and System Security (TISSEC), vol. 9, no. 2, pp. 162-180, May 2006.
- K. McKay, Trade-offs between Energy and Security in Wireless Networks Thesis, Worcester Polytechnic Institute, Apr. 2005.
- A. A. Tamimi, Performance Analysis of Data Encryption Algorithms, Retrieved Oct. 1, 2008. (http://www.cs.wustl.edu/~jain/cse567-06/ftp/encryption_perf/index.html).
- G. Ramesh and R. Umarani 'UMARAM: A novel fast encryption algorithm for data security in local area network' pp758 – 768
<http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=5654672>

- G. Ramesh and R. Umarani “Data Security in Local Area Network Based on Fast Encryption Algorithm” Communications in Computer and Information Science, 2010, Volume 89, Part 1, 11-26 <http://www.springerlink.com/content/m3301508219h7g66/>
- G. Ramesh and R. Umarani ‘A Novel Symmetrical Encryption Algorithm with High Security Based on Key Updating’ pp.57-69. <http://www.ijcns.com/pdf/207.pdf>

Appendix A
(S-BOX of Proposed Algorithm)

Row 1:86	Row 87:171	Row 172:256
6E 88 54 92 12 35 C9 D7	8D 15 28 66 A8 F3 55 B1	1C 06 D7 79 8A 83 73 D4
C7 58 26 E5 4B 96 C1 FF	0A 19 6A A8 11 9D A8 DF	5F E7 EE 75 0D B1 4B B5
0B 7D 0E 1F 6B FB 41 F1	17 70 BC F0 4D C3 01 16	A3 90 BE 0E A9 77 B2 E1
F8 E2 C9 59 81 EE 1C BA	7C E3 36 07 03 09 CB 90	A0 F4 C6 78 6D 64 8F 1A
FF F0 6E 0D E5 EA BB 33	6C 6E 06 27 A6 CE 7B 27	15 B4 A1 BA FD 0D 6D B7
75 D1 5D EA A7 7A 0E 8D	1D 41 8F 37 95 DD 78 11	C9 13 81 61 7F B7 31 21
C8 91 DE D0 D2 5A AB E0	4A C9 18 AA 9D BA D0 0A	56 78 AC DD A2 63 5D 6F
B9 89 FC 3C 59 D6 C0 08	21 B1 4B 88 66 05 29 24	92 AD 7A 2A 1C 0A C4 37
07 F8 66 9B 98 65 89 DA	E6 C3 8C 9E B7 86 53 B9	C1 DE 8E A9 51 CD 17 98
49 BF 59 31 45 22 EB CB	E8 2E 55 2E C9 27 FB AC	0E C6 3C 91 C1 D2 AF 1D
5C 1A B2 34 E7 34 35 57	B6 CA 42 FC 37 E2 E8 BF	AC 4E 4D C9 F2 72 BE 52
72 54 8B 55 3E 1D F4 02	2C C0 78 A3 B0 A1 D2 ED	96 7E 80 7C 17 24 4E 60
67 CC D8 4C 78 E3 E2 C8	88 29 DC 6D F8 CB FC 63	AF A1 B0 5E 84 1B F1 F7
FC 2C AF AC 04 A0 1A 7A	62 D2 B1 74 CA 2A C6 6D	2E 44 2D D2 F1 8A A2 8F
0F 05 FA 67 AD 02 30 B4	E5 A3 AE FB E4 DC F2 99	5E 4C 53 8E 52 F6 C2 07
C3 51 E7 65 36 A3 3B 0E	9E 52 08 52 0F 66 C8 28	03 FB 5C CD AF F4 2F FD
DF 39 6C C0 D1 1E 23 A8	81 46 65 5D 56 08 A5 DB	F0 04 9B 82 1D EF 48 E9
5B CF 48 57 2D 95 1B F2	14 FA 2F 8D 0E E6 E6 0C	91 27 AA 93 25 58 32 23
2D D7 1B 00 DB E5 37 FE	79 EF A7 99 F0 19 4A 87	A2 95 3F 97 77 6B 1E F8
45 45 52 45 A4 41 B0 22	ED 69 DA DA 76 7C 69 76	84 B5 9D EC C4 AD CE F3
E3 EA 51 32 9B 3C 79 67	7E 49 1C C4 C0 57 94 F5	60 4D CB 44 FA 9F 09 8C
FD 8E FB 4F 33 2B 56 AF	A6 E8 BD 02 CC E0 EA 93	7F 07 FE 6A 10 A6 61 EA
58 25 7D 25 ED F8 18 E3	EF A4 32 18 96 10 C7 5A	CD 01 27 B4 C3 53 2E B3
D2 B3 84 B6 75 9A A9 58	63 AF E9 AE 62 87 7A EE	A1 B9 7C EE 8E 36 7E 47
CC 16 99 8C 6C 5C 4F 36	32 4F F9 7F 54 04 5A 00	77 D3 58 C1 97 7B 5E 0F
FE DC 68 19 19 3E 57 A9	B1 FE F1 D7 7A 54 D8 1B	CA B2 40 BE 6A 88 87 73
AE 0D 22 7E 87 D0 0A D8	E7 77 38 76 28 1C 58 1E	94 A9 94 38 79 A8 77 E2
6F 11 B7 E7 82 12 8C 7F	8A 23 9F 53 7E 98 85 CE	53 63 25 46 94 AE B4 84
A8 03 70 B2 3D 6A 92 AB	36 92 B5 30 EF EB 49 0B	3D 6C 87 20 4C F5 52 01
B4 D5 0B BB 4E E7 36 5B	C4 AB 20 83 CB 7D CD 2C	0C 65 D6 81 E9 DB DD 06
70 BD 10 D8 AA 8E 33 9C	22 80 49 7B 63 4F 86 83	3C 00 A8 5F B5 03 5C B0
80 93 2C D1 02 8B EE 9E	A5 2B D2 17 AC CC CA 05	8F 75 4C 63 43 FC 60 C4
34 82 33 35 C7 73 CC F9	69 3C 93 69 70 E1 2D 3A	09 59 62 73 EB FF FF E7
B0 A0 EA 95 FB 4C D7 5D	87 C5 AD BD 49 82 70 C3	93 20 75 6B DF 48 0C 18
13 F5 09 28 39 1A 26 74	E1 5B 24 0C AE 61 B1 96	25 73 B3 FA BA 43 9E C1
5D 10 7F E3 F9 50 71 55	26 14 E2 C5 AB 30 D6 68	71 BE 5A 29 9E 81 E1 8A
BC 8D B6 39 D0 78 2B D6	0D 6D 05 6F 50 B2 4C E5	31 87 92 CF CF 4E 11 2F
7B 96 A3 BC 16 C6 16 CD	55 5A C3 8F 74 31 9C 25	50 5C 1A 24 06 26 90 30
F6 22 D0 A2 9A 01 E0 79	29 E4 69 4A B6 C2 E9 59	E2 83 97 10 1F 93 05 42
5A 30 5F 3F 5A C1 BC F0	A9 A2 73 3B 48 16 02 8E	DB 61 B4 06 B8 75 63 17
65 85 46 70 22 56 83 A4	89 C2 EF 71 1A E4 B6 6A	41 4A A4 40 80 2C C3 40
2F 5D 35 EB CE A7 62 7C	1B A7 E6 B0 6E 23 A6 C0	AB 97 E8 8B D5 FD E5 54
F2 EE 63 8A B2 94 47 FA	97 8C 9E D3 A5 B6 74 1F	AA F7 82 D4 41 39 1D 86
44 24 15 2D E1 2F 75 32	38 9F CE 03 C8 B0 F6 66	24 79 07 41 89 C8 A7 2E
73 34 D3 A0 34 1F 0D 61	F3 F2 77 1B BB D5 20 9F	00 21 A6 DF 6F AB 12 EB
EB 76 A9 11 2B B8 A0 12	74 28 AB E0 73 49 D1 A2	1A 6F F0 47 1E 0C ED 3E
D6 C1 E4 EF 5F A2 5F B8	C6 42 DD B7 47 17 3D BE	54 DB 45 85 EE 45 38 A3
CF BA 2B F4 53 67 3F 29	EE 71 F8 F2 D9 DA 28 4F	08 3F 85 42 69 69 9D 3D
D0 0E 79 A7 72 3B 27 92	D5 74 5E 98 F6 3D EC FC	66 36 C8 12 DD E8 D9 D2
3F B0 7B 96 09 4D 43 CF	68 C4 6D 1A 99 C7 CF 77	E0 EC 64 9F 4F 62 19 48

2B BC F5 F6 14 6C 50 6E	4F 94 C4 9D 5D 97 00 62	99 C8 5B DB B1 84 A1 9F
04 1E 34 15 E8 20 8E 45	40 08 9C D6 3C F2 07 7E	9B 43 9A E1 EA 06 B3 4D
A7 E1 0A 2C 0A 33 2C 20	CE ED B8 AB 2A 8D E3 71	57 F1 91 CC D7 9E 95 10
EC 0A 00 FF 20 29 40 50	98 3E 1F E2 D4 0B 42 39	06 26 6B E9 83 37 FD DC
18 1D C2 0F 3B 59 DB 72	BF 1B 4E FE 18 5E 80 EC	AD CE 76 2B DE 4A 34 78
B2 12 E0 60 5B E9 51 95	05 FF 8A 50 67 38 24 56	C5 0C 60 90 8B 79 03 D1
39 D9 8D 3D C5 70 39 81	7D 33 1E C2 B9 B3 15 44	16 8A 0C 3E 23 A9 59 14
8B 38 D5 22 24 FF 10 38	B8 AE F3 9C 7B 5B 3A CA	FA FD 7E 64 9F 2E 7C 9A
76 8F 43 43 4A 47 9A 46	59 D0 4A A5 E2 F1 AD 3C	7A D4 12 04 01 F7 45 64
EA 0F 30 36 7D 92 96 41	95 F3 50 F5 B4 DF 06 BC	3E 67 67 F3 55 7F 9F 5E
90 3D 14 89 07 B4 76 C9	30 1C 3B 94 00 CF 72 B2	CB E9 1D 0A 1B 0F AC 70
DA 84 86 1C 38 4B 3C C7	6B 40 D1 4D BC 76 1F CC	FB 60 DB 33 F4 D4 6B 5F
D8 50 19 0B E3 46 8B E6	43 6A 2A C6 92 C5 88 C5	BB 7C 37 C8 E0 6D E7 9D
F9 2F F2 F9 27 BD DA 31	37 DF 11 5B 21 ED BD 91	C0 6B 04 DE 05 21 14 3B
D3 35 EC A6 D3 C4 93 89	78 3B 01 6C 31 42 25 B6	D7 09 72 CB 29 44 F9 2D
3B 02 13 D5 D6 A4 5B D3	82 55 88 C3 BD D1 D4 80	8C 86 A0 51 BE 40 DC 35
D1 8B 96 F8 DA 60 84 51	B3 F6 71 08 86 55 EF 8B	61 D8 95 16 F5 9B 8A 69
F4 31 CD CA 3A 3A DE 94	64 A5 4F 87 68 18 9B DD	6D 1F 03 D9 C2 AC 2A 85
8E 47 EB 4E BF AA B8 2B	E4 3A 17 26 9C DE F3 49	4E 98 BB 72 2E 7E 68 6C
33 E5 D9 09 13 D9 13 13	F7 99 ED 14 08 BB F7 DE	27 48 A5 68 58 D8 04 BB
3A 53 3E A4 C6 28 66 A0	1E 7B F4 9A 8F 89 F5 4E	02 B7 90 B8 F7 6F DF 53
C2 2A 3A 84 D8 A5 44 FB	E9 DA 57 5C 88 BE 4D 4B	11 64 89 B5 85 3F 82 2A
DC BB CC 7D FC 11 F0 A6	1F CD DF CE 5E 80 FE 65	BE 9A 56 54 E6 8F 91 82
9A 72 83 77 A1 74 6A 09	DD E6 BF E4 DC 71 81 F4	9F 0B BA E8 2F 9C AE 43
4D 68 E1 3A 7C 90 D3 5C	46 2D 0D FD 2C EC 21 AE	6A 9C C7 ED 91 32 22 A7
9D AC 3D 01 FF 5D AA 4C	4C DD 61 49 FE AF B7 C2	51 81 B9 86 32 52 98 26
4B 7A 21 C7 0C CA B9 A1	83 B8 2E B1 26 51 7D BD	B7 9B E5 F1 A3 F9 D5 9B
DE 9D 31 58 93 B5 BF 4A	D9 FC 0F F7 61 25 64 AD	BD AA 16 62 42 85 08 03
9C B6 CF 1D 44 8C B5 EF	28 A6 41 13 46 D3 6C AA	20 EB F7 23 3F 6E 54 3F
85 18 74 DC 15 99 6E 88	52 66 47 80 90 F0 99 75	2A A8 FF BF CD 0E 67 E4
35 32 23 7A 8D 91 3E D0	48 7F C0 5A 40 D7 A4 6B	47 5F 02 21 64 15 BA E8
42 37 A2 B3 EC 14 0F F6	F5 D6 29 05 30 68 7F 34	86 62 CA 48 60 07 A3 0D
B5 C7 F6 B9 65 13 65 A5	10 57 C5 56 57 C9 FA 1C	BA E0 D4 AF A0 FE E4 15
F1 5E C1 2F 71 B9 46 C6	19 17 E3 AD B3 2D C5 7D	12 9E FD A1 8C C0 6F 19
D4 F9 6F 6E F3 00 0B D9	A4 4B 44 1E 5C BC F8 D5	01 CB 39 4B 35 FA 8D 04
23 56 98 E6 0B 5F 97 7B		

