Journal of Theoretical and Applied Information Technology
15th February 2020. Vol.98. No 03

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

THE NEED FOR POLYMORPHIC ENCRYPTION ALGORITHMS: A REVIEW PAPER

[1]ABDELRAHMAN ALTIGANI, [2]SHAFAATUNNUR HASAN, [3]BAZARA BARRY
[1,2]School of Computing, Faculty of Engineering, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

[3]Faculty of Mathematical Sciences, University of Khartoum, 11111, Khartoum, Sudan

E-mail: [1]a.altigani@gmail.com, [2]shafaatunnur@utm.my, [3]bazara.barry@outlook.com

ABSTRACT

Current symmetric ciphers including the Advanced Encryption Standard (AES) are deterministic and open. Using standard ciphers is necessary for interoperability. However, it gives the potential opponent significant leverage, as it facilitates all the knowledge and time he needs to design effective attacks. In this review paper, we highlight prominent contributions in the field of symmetric encryption. Furthermore, we shed light on some contributions that aim at mitigating potential threats when using standard symmetric ciphers. Furthermore, we highlight the need for more practical contributions in the direction of polymorphic or multi-shape ciphers.

Keywords: *Cryptography, Polymorphic Cipher, Advanced Encryption Standard, Symmetric Encryption Algorithms, Multi-Shape Cipher.*

1. INTRODUCTION

The convenience and low cost of deploying and using online services have dramatically increased the utilization of E-Services as a complementary or even a replacement for traditional and manual services' ports. In [1], it has been reported that ICT increased labor productivity by at least 31% in the European Union and by 33% in the United States since 1995.

Unfortunately, a significant segment of society is still reluctant to utilize E-Services. This is mainly due to the widespread of cybercrimes or specifically, the cybercrimes which target consumers (consumer-oriented cybercrimes) [2]. As stated in [3], although the direct effect of cybercrimes is significant, the indirect cost is even higher, as many potential users avoid cyber precarious scenarios by simply avoid using the E-Services. Therefore, unless E-Services have been equipped with proper and reliable security services, E-Services cannot be fully utilized.

Each security service can be provided using the appropriate mechanism or mechanisms. For instance, to provide the Confidentiality security service, we usually use encryption. Other security services, such as Integrity, or Availability can be provided using other cryptographic mechanisms.

The next section will introduce the building blocks for symmetric encryption. This includes a brief background for symmetric encryption, as well as introducing some of the well-known symmetric ciphers including DES, 3DES, Serpent, Twofish, RC6, MARS, AES. After that, we shall highlight some of the contributions in the direction of using non-standard ciphers. This includes AES-Like ciphers, Steganography, Split and Encrypt, Encryption Cascading, Chaotic Encryption, Dynamic Encryption among other ciphers. Consequently, we have discussed a few encryption techniques which have been designed to fulfill the requirements of a specific environment. Finally, a brief contribution is provided to stress on the contribution of this research paper, and the need for polymorphic ciphers.

2. CRYPTOGRAPHY

The need to keep some sort of communications secret has been realized early in history. For instance, it has been discovered in the remnants of an ancient Egyptian town, 4,000 years old hieroglyphic inscriptions written in an unusual approach to obscure its meaning [4]. However, the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

last century has seen a dramatic peak in the evolution of encryption algorithms. Perhaps, the pivotal reasons were the world wars, and later in the century, the widespread of computer networks.

Encryption algorithms scramble the secret data into an unintelligent form. The scrambling process is called encryption or encipherment. Only the legitimate receiver can recover the encrypted message (i.e. decrypt or decipher the message) using a data structure called the encryption key. This key is typically kept privately with the legitimate receiver [5-7].

Encryption algorithms can be classified according to three different criteria [8]:

1. The operations used to encipher the plaintext.

2. The number of used keys.

3. How the plaintext is processed.

However, a slightly less formal, yet widely used, classification focuses only on the second criterion, the number of keys. According to the latter classification, encryption algorithms can be classified into symmetric and asymmetric encryption algorithms.

Symmetric encryption algorithms use the same key for both encryption and decryption. Therefore, communicating parties are expected to "somehow" share the encryption key. This is the main limitation of symmetric encryption algorithms which is also known as the key exchange problem [9, 10].

On the other hand, asymmetric encryption algorithms use two different keys per user, a public key and a private key. For instance, if Bob wishes to send a confidential image to Alice, he will fetch Alice's public key and encrypt the confidential image using her public key. Consequently, only the holder of the corresponding private key (i.e. Alice) will be able to decrypt the image using her private key. Alice's public key is generally posted in an integrity-protected format and is accessible virtually to everyone.

Although public-key ciphers resolved the key exchange problem, they are considerably sluggish compared to symmetric ciphers. This is why most contemporary cryptographic protocols use a hybrid of both symmetric and asymmetric ciphers to utilize the virtues and eliminate the drawbacks of both schemes.

This research paper focuses on symmetric encryption. Hence, the following subsections highlight the prominent contributions on this area, and guide the reader to a potential research area.

2.1. Data Encryption Algorithm

Numerous ciphers are available. However, to facilitate confidential communication across various platforms, intuitively, the communicating parties must agree on the used encryption algorithm. Therefore, in 1977, the National Bureau of Standards (Currently known as the NIST) has selected the Data Encryption Algorithm (DEA) as the standard cipher [8]. Until the introduction of the Advanced Encryption Standard in 2001, the DEA was the most widely used cipher [8]. It also worth mentioning that the terms Data Encryption Algorithm (DEA) and the Data Encryption Standard (DES) are used interchangeably in the literature.

In the DES cipher, the length of the block size is 64 bits and the length of the key is 56 bits. A dedicated module is used to expand the key to 16 sub-keys. Each of these sub-keys has 48 bits. As will be described shortly, every sub-key is used in one round. DES has 16 identical rounds which are used to transform the plaintext into the corresponding ciphertext. Moreover, DES cipher has a Feistel structure. This means, in every round, only 32 bits of the input is processed by the encryption function. Figure 1 depicts the Feistel structure of an arbitrary round in the DES.

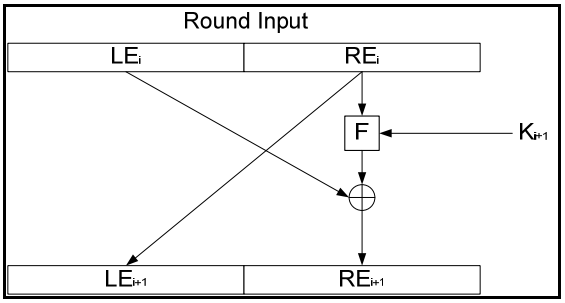

Figure 1: Explaining the Feistel Structure (Stallings, 2016)

The F function receives a 32 bits input and produces a 32 bits output. It includes an expansion operation, an X-OR operation, a substitution operation, and a permutation operation. The expansion operation is carried using a dedicated data structure called the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Expansion Box (E-Box). This operation expands the received 32 bits to a corresponding 48 output bits. The 48 output bits undergo an XOR operation with the 48 bits round-key. The resulting 48 bits are split into 8 blocks. Every block is fed to one Substitution Box (S-Box) of the eight available S-Boxes. These S-Boxes are usually denoted as S_1, S_2, S_3, S_4, S_5, S_6, S_7 and S_8. The DES designers were aware that these S-Boxes are the only nonlinear element in the cipher design. Hence, it has been selected with caution to maximize the confusion property in the DES design. The last step in the F function is a permutation operation that provides a layer of diffusion. This is because the 4 bits output of every S-Box will be scattered within the 32 bits. Consequently, it will be substituted by different S-Boxes in the subsequent rounds [11].

In 1994, the NIST realized that given the rise in the computational power, the 56 bits key length of the DES cipher is not going to be sufficient to counter brute force attacks in the future. Hence, in that year, the NIST declared that DES can be used for five more years to secure unclassified federal communications. In 1998, the Electronic Frontier Foundation managed to develop hardware called DES cracker that successfully retrieved the DES key in less than three days [12-14]. This experiment has proved that, the DES cannot resist brute force attacks [15]. Therefore, in 1999, the NIST declared that the DES cipher should not be used except for legacy systems. They also introduced the Triple Data Encryption Algorithm (TDEA), which is commonly known as the Triple-DES or 3DES for short [8]. The next section will highlight some aspects related to the 3DES cipher.

2.2. Triple Data Encryption Algorithm

The Triple Data Encryption Algorithm simply applies the DES algorithm three times. Assume M is the message that needs encryption, E is the encryption module of the DES, and D is the decryption module of the DES. TDEA works as specified in equation 1:

$$C=E_{K3}(D_{K2}(E_{K1}(M))) \quad (1)$$

Where C is the ciphered version of the message M. Moreover, K_1, K_2, and K_3 are three different DES keys. This arrangement increases the key size of the key size from 56 bits to 112. Note that the key length is not multiplied in three as expected. This is due to the meet-in-the-middle attack, which significantly reduces the effective key bits [8, 16].

TDEA inherits the strength of the DES and resolved the short key problem realized in the DES algorithm. However, TDEA is a sluggish algorithm, which has a relatively poor performance.

2.3. Advanced Encryption Standard Competition

2.3.1. Introduction

In addition to the short key of the DES cipher, several cryptographers have always criticized the "closed-door" strategy adopted by the National Bureau of Standards in selecting the standard cipher. Therefore, in 1997, the NIST has published a Call for Proposal that request interested cryptographers to design new ciphers that can replace the DES and TDEA. The winning algorithm will be the called Advanced Encryption Standard (AES). The NIST has specified a number of evaluation criteria which can be summarized in three groups: security, cost and implementation characteristics. Furthermore, the NIST has specified a number of guidelines to which all submissions must abide. For instance, the cipher must be a block cipher with a block size of 128 bits. Moreover, the cipher must support the key lengths 128, 192 and 256 bits.

Initially, the NIST received 21 candidate algorithms. Using the aforementioned criteria, this number has been reduced to fifteen ciphers and in the next round, it has been reduced to five ciphers. These ciphers are Rijndael, Serpent, MARS, Twofish, and RC6. All these five algorithms are outstanding. Nevertheless, Rijndael has been selected as the Advanced Encryption Standard (AES). In the following subsections, we shall provide a brief description for each one of these five ciphers.

2.3.2. Serpent

During the AES competition, Serpent cipher managed to reach the final round against Rijndael cipher, where it gets 59 votes against 86 votes for Rijndael [17, 18]. The Serpent has 32 rounds, regardless of the length of the key. Every round contains a key-mixing, byte substitution and a linear transformation to provide diffusion. Due to the large number of rounds, Serpent is known to have a high-security margin. Nevertheless, this huge number of rounds degrades the cipher's performance.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

According to Serpent authors, one of their key design strategies is to inherit the strength of the DES cipher, yet promoting the cipher performance. Therefore, Serpent uses the same DES S-Boxes. For every round $(round)_i$, a particular S-Box $(S\text{-}Box)_i$ is used. Moreover, in every round, Serpent makes 32 replicas for $(S\text{-}Box)_i$. The 128 bits input to this round is sliced to 32 groups of four bits. Every four bits group uses its own replica of $(S\text{-}Box)_i$. Consequently, for all 32 groups, the substitution process is parallelized to optimize the cipher performance [19].

2.3.3. RC6

This cipher is introduced by Rivest, Robshaw, Sidney, and Yin as a candidate for the AES competition. The RC6 design is mainly an improvement to the RC5 cipher to fit the requirements of the AES competition. Moreover, compared to the RC5 cipher, several optimizations have been incorporated such as including a multiplication phase on the RC6 design to provide faster diffusion, decreased number of rounds, increased throughput, and increased security. It worth mentioning that RC6 is not a royalty-free cipher.

RC6 is a flexible cipher, which can be parameterized in a number of different ways. However, like all other candidates, the RC6 can support a block size of 128 bits. Moreover, RC6 has a Feistel structure and the default number of rounds is 20 [20]. During the AES competition, RC6 managed to secure 23 votes [18].

2.3.4. Twofish

Twofish is a Feistel-like cipher that has 16 rounds. It has been described as a Feistel-like cipher because some parts of the round input are rotated before being processed by the F function, where F is the function responsible for carrying out the encryption transformations. Hence, because not all transformations are localized inside the F function, it is called a Feistel-like cipher.

The F function includes byte substitution using four different S-boxes. These S-boxes are key-dependent. It also includes a multiplication in the matrix under $GF(2^8)$. The aforementioned operations are carried inside the F function by the sub-function g. Inside the F function, the g function has two replicas. Each one of the g function replicas receives and processes one word (i.e. four bytes), and outputs one word. The resulting two words are then processed with each other using a Pseudo-Hadamard Transformation (PHT). The two words, which comes out of the PHT transformation, are added to specific words of the round key. At this point, the F Function has finished its processing for this round. However, before proceeding to the next round, the resulting words are rotated. Then, the Feistel structure is invoked to XOR the processed half of the input with the other half. Consequently, the two halves are swapped [21].

It may worth mentioning that in the AES competition, Twofish has secured 31 votes [18]. That makes Twofish cipher in the third place after Rijndael and Serpent Ciphers [18].

2.3.5. MARS

This cipher proposal was submitted by IBM©. It worth mentioning that IBM is one of the big names in the cryptography field. This is because they have introduced a cipher upon which the DES cipher has been standardized by the National Bureau of Standards in 1977 [22].

Like all other AES candidates, the block size of the MARS cipher is 128. Furthermore, MARS supports all the key lengths 128, 192 and 256 as specified by the NIST. However, it can also support other key lengths in the range 128 to 1248 bits. As justified by its authors, supporting keys longer than 256 is mainly to fit some existing key-sharing protocols, such as Diffie-Hellman, which usually shares a symmetric key of lengths beyond 256 bits.

The cipher has a Feistel structure with 32 rounds. Before we commence processing the data using these rounds, a key addition (modulo 32) is performed between the plaintext and one of the keys generated from the expanded key. Similarly, after finishing the 32 rounds, a key subtraction is carried (also modulo 32) between the processed data and one of the keys generated from the expanded key. The first 8 rounds, as well as the last eight rounds, are unkeyed. These sixteen rounds contain mixing operations, and byte substitution using two S-Boxes consisting of 256 words, where every word is 32-bits. The remaining sixteen rounds contain keyed transformations which include multiplication, rotations and S-Box lookup [23].

Not as expected, the MARS cipher gets only thirteen votes [18].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

2.3.6. Rijndael

Rijndael is a symmetric block cipher. The size of the block can be 128, 192 and 256 bits. The key can have the lengths 128,192, or 256 bits. In the start, the cipher splits the input to *n* blocks according to the desired block size. If necessary, a padding scheme is invoked to fill the remaining empty bytes of the last block. The first block is copied in plain into a matrix called the state matrix. The state is always modified until reaching the final encrypted text. The structure of the AES is not Feistel. This means all the bits in every data block are processed in parallel during each round using substitutions and permutations.

The key is expanded to be $4 \times (N_r + 1)$ words, where the word consists of 4 bytes. N_r represents the number of rounds, which is a function of the key size and block size. Table 1 depicts the value of N_r according to the values of the key size and the block size:

Table 1 Rijndael Number of Rounds as a Function of the Key Size and the Block Size

Block Size / Key Size	16 bytes	24 bytes	32 bytes
16 bytes	10	12	14
24 bytes	12	12	14
32 bytes	14	14	14

At every round (except the final round), we have four stages (or layers), which are:

1. ByteSub.
2. ShiftRow.
3. MixColumn.
4. AddRoundKey.

The encryption process starts with an initial AddRoundKey operation, followed by $(N_r - 1)$ identical rounds. Each one of these rounds consists of the above four layers. After that, a final round is applied. This final round consists of all these layers except the MixColumn layer.

Each one of these layers has its own function. ByteSub maps each byte in the state to another value determined by the S-box. The S-box can be viewed as a 16×16 matrix, with all the possible bytes values'. This process can be inverted using the inverse S-box. This operation maintains a low correlation between input bits and output bits. ShiftRow shifts the last three rows in the state cyclically to the left by different amounts. After completing this operation, it is guaranteed that every 4 bytes which were previously on the same column are currently on different four columns in the new state. MixColumn intends to add confusion. To get the output from this layer, we have to multiply our state matrix, by the following matrix:

$$\begin{bmatrix} 02 & 03 & 01 & 01 \\ 01 & 02 & 03 & 01 \\ 01 & 01 & 02 & 03 \\ 03 & 01 & 01 & 02 \end{bmatrix}$$

During multiplication, the state matrix should be to the right, and the above matrix should be to the left. Each new value in the new state (i.e. the state after the MixColumn operation) is identified by the summation of the values of all the column elements, multiplied by some value, specified according to the corresponding value in the above matrix. It is worth mentioning that the multiplication process is performed under Galois Field $GF(2^8)$, to guarantee that the outcome length will not exceed 8 bits.

Finally, the AddRoundKey operation is a simple XOR process between the bytes of the state and the bytes of the round key. The latter operation equips our state with initial security, while the other layers add extra complexity. Rijndael has won the AES competition by securing 86 votes.

2.3.7. Selecting the AES

In contrast to the DES cipher, the process of selecting the AES has been open to feedback from virtually everyone. Hence, after eliminating all unsuitable candidates, cryptographers from all over the world were approached to provide their feedback regarding these five candidates. As aforementioned in the previous subsection, the number of votes secured by the Rijndael cipher exceeded all other candidates. This is because, among all five ciphers, Rijndael is thought to provide the best trade-off

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

between security, performance, and implementation cost.

After selecting Rijndael, the NIST has published the Federal Information Processing Standards Publications 197, commonly known as FIPS 197, declaring that Rijndael as the Advanced Encryption Standard (AES). However, in this publication, they have introduced slight changes to the names of the AES operations. For instance, the ByteSub has been changed to SubBytes. MixColumn and ShiftRow have been changed to MixColumns and ShiftRows respectively. Moreover, although the Rijndael cipher can support three block sizes, it has been fixed to 128 bits in the NIST specifications for the AES.

2.3.8. Concerns when using the Advanced Encryption Standard

The reader should know that the only way to determine whether an algorithm is secure or not is by assuring that the cipher can resist known attacks such as differential and linear cryptanalysis. Things are always changing, and it is possible that a new technique has been discovered, yet has not been publically known. This has been seen in the OpenSSL implementation (i.e. the Heartbleed bug). Needless to say that the same scenario can be repeated in the design or implementation of any encryption protocol or encryption algorithm. The current trend of standardizing and leaving the exact operation details of the encryption algorithm open and accessible by everyone is crucial for interoperability. However, it exposes the algorithm to significant hazards. Moreover, it has been suspected that standardization agencies have some undisclosed agenda for encouraging the use of an open and standardized algorithms. This concern has stated explicitly by some researchers [24].

The active research work in designing and introducing new symmetric encryption algorithms (not to mention the classified work in some government agencies) acknowledges the implicit consensus among a wide segment of cryptographers to the need to avoid using the AES.

Furthermore, many attacks are designed to penetrate the AES [25-34]. Although they are still considered impractical, they have achieved partial success. Probably it is only a matter of time until an effective attack will be publically published.

It can be concluded that it is generally safer to be prepared for the possibility that the AES can be penetrated, rather than trying to patch the problem when it has already occurred.

If we can keep our algorithm design open (to preserve the interoperability and facilitate testing), yet eliminate the opponent knowledge about the exact used encryption algorithm and other cryptographic parameters, that will significantly enhance the security of the used encryption algorithm.

3. EXAMPLES FOR ENCRYPTION ALGORITHMS AND ENCRYPTION PROTOCOLS PROPOSED TO AVOID USING THE STANDARD SYMMETRIC ENCRYPTION ALGORITHM

This section highlights alternative ciphers that have been suggested to avoid using the symmetric standard encryption algorithm, namely the AES. The active research work acknowledges the implicit consensus for not to rely entirely on the AES as an ultimate solution for providing confidentiality.

3.1. Combined Encryption

The impressive study of C. E. Shannon [35] explored the possibility of combining encryption algorithms or "secrecy systems" in a variety of ways. Assume we have the encryption algorithms T and R. A new algorithm S can be developed as the "product" of the algorithms T and R. Thus, $S = T.R$. This means the data is encrypted by both T and R (i.e. encryption cascading). Another way to combine the encryption algorithms is "weighted addition". This means assigning a probability for using each algorithm. Therefore, $S = pT + qR$, where p is the probability of using T and q is the probability of using R. Needless to say that $p + q = 1$. This means with every different message, S encrypts using either T or R according to the probability given to each encryption algorithm.

Both models can be generalized to include n different encryption algorithms. For instance, weighted addition may incorporate n different encryption algorithms as $S = \sum_{i=1}^{n} p_i C_i$, where $p_i$is the probability of using the encryption algorithm C_i, and of course $\sum_{i=1}^{n} p_i = 1$.

Although encryption cascading protocols have been introduced and closely studied [36, 37],

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the performance degradation associated with encryption cascading has always been an obstacle. On the other hand, alternating between several algorithms did not receive enough attention. One possible reason is that there is no need to use a set of algorithms when we can directly use the best algorithm in the set. Implementing and managing a group of algorithms adds unnecessary complexity to the encryption model design without any tangible benefit.

Few other studies have discussed the concept of combined encryption algorithms. However, most of the proposed models are still deterministic and open. This means communicating parties, as well as anyone in the middle, can see the design and operation details of the used encryption algorithm.

An example of the combined encryption algorithms is TDEA. Although TDEA is the most well-known example of encryption cascading, additional models exist. For example, in [36] the authors proposed a model in which AES, Serpent and Twofish encrypt the message successively (see figure 2). Initially, three different keys are generated on the sender side, one key per algorithm. The data are then encrypted successively three times by different key and a different algorithm from the above list. The keys (i.e. symmetric keys used with the AES, Serpent, and Twofish) are encrypted by RSA using the recipient's public key. The encrypted keys are attached to the beginning of the encrypted message. On the recipient's side, the keys are decrypted using the recipient's private key. Then, each key is used along with its corresponding algorithm to decrypt the message sequentially until we retrieve the plaintext.

The logic of this model is that the message will be secure as long as at least one of the three mentioned algorithms is secure. Furthermore, the key length increases significantly, as three different keys are used instead of only one.

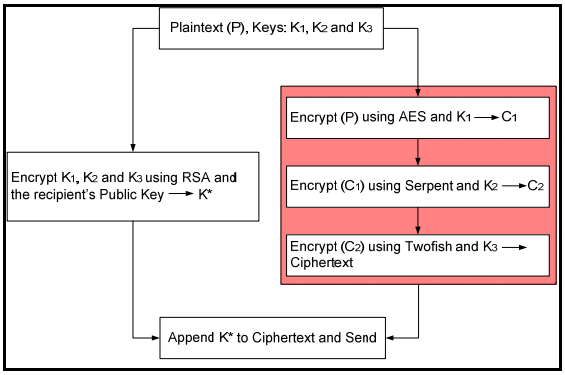

Figure 2 Multilayer Hybrid Encryption [36]

Although no clear information has been provided regarding the performance of this model, it can be easily inferred that the required computational overhead, due to applying all these encryptions will not be tolerable for most applications.

3.2. Dynamic Encryption

Although it is less common, a few researchers had some attempts on the direction of dynamic encryption. For instance, C. B. Roellgen has introduced a cipher that has eight different base ciphers. These ciphers are Rijndael, Serpent, Twofish, Anubis, RC6, SEED, Serpent and Cast-256. Depending on the key value, the cipher will suggest a specific sequence for executing eight ciphers out of the above eight base ciphers. This sequence can be the same cipher repeated eight times (e.g. $serpent > serpent > serpent > serpent > serpent > serpent > serpent > serpent$) or all the eight ciphers, where every cipher appears just only once. The number of possible permutations is 8^8 which is equal to 16,777,216 different possible permutations. The plaintext will be encrypted using all the ciphers specified in the randomly selected sequence of permutation and the same order [38].

Although the encryption time is not provided for this new cipher, it can be easily guessed that the performance will be severely degraded. This is because now we shall apply 8 different ciphers for encrypting any given plaintext, which is a computationally intensive procedure. Needless to say that high performance is a crucial requirement for most platforms. Hence, poor performance cipher will not attract the attention of developers.

Another work that has been introduced by Ijaz Shoukat [39] is a suitable example for a fully

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

dynamic cipher. In this work instead of processing fixed-size blocks, they have proposed a Dynamic Data Blocking Mechanism (DDBM) to generate dynamic sized data blocks. Furthermore, a Randomized Substitution Mechanism (RSM) is introduced to unpredictably alter the keys and the blocks of the plaintext. After that, a Multi Operation Data Encryption Mechanism (MODEM) is used. The MODEM operates by dynamically picking a group of encryption operations among several other groups. Each group includes a set of operations such as XOR, permutations, random substitution, shifting, and logical operations. The process of selecting which group to be used is key-dependent.

The main criticism to this model is that it has quite poor performance. To illustrate, the time needed to encrypt 13216 bits ($\approx 1.6\, Kilo\, Byte$) using this algorithm is 21.21 seconds, and the time needed to decrypt the same amount of bits is 124.49 seconds. Hence, it is fair to say that this model is impractical.

Another dynamic encryption approach is based on DES and matrices multiplication. In this model, the plaintext x is initially multiplied in a binary invertible matrix k_a. k_a is generated according to the concepts of Network Coding and using an arbitrary integer positive value D_a. After carrying out the multiplication process, x will be transformed to z_1. Consequently, the DES cipher is invoked to encrypt z_1. The outcome of this step is z_2. According to the authors' statement, the main reason for invoking the DES cipher is to "bring non-linearity". After that, another binary invertible matrix k_c is generated and z_2 is multiplied in k_c to get our ciphertext y. The details of generating k_a, k_c as well as the routine used to update k_c is elaborated in [40].

An essential step suggested in the aforementioned study is to update the matrix k_c before sending new messages. Consequently, even if the same message has been sent twice using this model, the outcome will be different, even before employing any block cipher mode of operation. This change leads to dynamic cipher. The process of updating k_c is called a partial key update. This is because the 64 bits DES key, k_a and k_c are together used as a key for this new cipher.

According to the statement of the authors, the performance of this model is comparable to the 3-DES cipher. Nevertheless, 3-DES performance was never considered acceptable. In fact, one of the main objectives for the AES competition is to overcome the sluggishness of the 3-DES cipher [8]. Moreover, the choice of the DES cipher in the intermediate layer is hard to understand, especially when you have other secure and efficient alternatives. Furthermore, according to the statement of this cipher authors, the security of this model needs further analysis, which has been postponed as future work. Hence, it is early to assure the strength of this cipher in terms of security.

Another example of a dynamic encryption algorithm has been introduced in [41]. It may worth noting that this cipher also belongs to the family of lightweight ciphers. In this algorithm, the main concern is to drastically enhance the performance of the cipher by reducing the number of rounds to only one round. This is because, according to their claim, several delay-sensitive applications cannot tolerate the delay introduced by typical ciphers including the AES. On the other hand, the cipher must maintain adequate security levels to resist all known attacks.

It is assumed that the communicating parties have exchanged a secret Session Key (SK) a priori of establishing their communication. Using SK, an XOR operation is carried out with a 512 bits nonce. The resulting 512 bits are hashed using SHA-512. The result will also be 512 bits. These bits will change with any new nonce, hence it will be called the Dynamic Key (DK). DK is divided into 5 sub-keys: $\{k_{S1}, k_{S2}, k_P, k_{RK}, k_{SRK}\}$. k_{S1} and k_{S2} are used to construct 2 different key-dependent substitution table S_1 and S_2 using the key setup algorithm of the RC4 cipher. k_P is used to construct a permutation table π. k_{RK} will seed a stream cipher to generate a random sequence of bits. These randomly generated bits are divided into m blocks, where m represents the number of blocks of the plaintext. Every block of these m blocks will be used as a sub-key to be XORed with the one block in the plaintext. The k_{SRK} will be used to generate a selection table that specifies which sub-key to be XORed with which plaintext block. All the cipher's building blocks are key-dependent. Hence, any change in any part of the key will lead to a major and unpredictable change in the output.

The cipher processes two blocks at a time. The first block is XORed with a sub-key selected using the selection table. Consequently, the result undergoes a byte substitution process using S_1. The outcome of this substitution is XORed with the second plain block and the result undergoes a byte substitution process using S_2. The result is the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

ciphered version of the first block. The second plain block undergoes a slightly different process [41].

The authors of the latter approach claim that the cipher demonstrates adequate security level in resisting all attacks including linear and differential attacks. However, it is known that the repetition of the transformations of any cipher in several rounds is the only approach to decrease propagation ratio and correlation for linear trails. However, there is no compelling argument to support the claim of resisting linear or differential analysis.

3.3. Split and Encrypt

Instead of cascading encryption algorithms, in [42] they chose to split the plaintext into different pieces $p_1, p_2, \dots, p_n$ of equivalent length. Each peace p_i is passed to a different crypto-processor (i.e. encryption process) EE_i, which may apply a different encryption algorithm. It is necessary to ensure that the first piece of the input (e.g. p_i) has been encrypted and ready for transmission before the second peace (p_{i+1}), and so forth. This, however, is not always true, as the encryption burst depends on the base ciphers' performances. Therefore, we might have a case in which c_{i+1} (i.e. the result of $EE_{i+1}(p_{i+1})$) is ready for transmission, but c_i (i.e. $EE_i(p_i)$) is not ready yet. This results in ciphertext collision or disordering which might render the whole encryption protocol impractical. To avoid such outcomes, the maximum possible encryption burst is identified (the critical encryption burst). For other EE_i, a delay is added to make the encryption bursts for all the pieces of the input equal. Consequently, the order is preserved.

In this model, the identity of the algorithm which has processed each part is declared. Thus, if some of the input pieces have been encrypted using a weak algorithm that can be penetrated, all these pieces will be subject to effective cryptanalysis attacks.

3.4. AES like Encryption

Some researchers suggested using modified versions of the AES. The logic of this set of new algorithms is to introduce a minor and secret change in the design of the AES. This results in a new encryption algorithm with the inherited strength of the AES. Changes might include using a dynamic S-Box as proposed in many studies including [43-47]. Nevertheless, it can be argued that it is not judicious to manipulate the AES S-Box. This is because the AES S-Box has been selected with caution to keep the maximum values of prop-ratio and input-output correlation as minimal as possible. Consequently, the prop-ratio and input-output correlation of the AES S-Box are less than 2^{-6} and 2^{-3} respectively. These values play an essential role in determining the resistance of the cipher to differential and linear attacks. Using other S-Boxes from the space of 8-bits invertible S-Boxes will typically have the scores 2^{-5} to 2^{-4} for the maximum prop ratio and 2^{-2} for the maximum input-output correlation [48]. Consequently, the overall resistance of the cipher against linear and differential attacks will decrease.

ShiftRows transformation can also be updated (e.g. shift the row i in the state matrix by x_i, where x_i can be 0, 1, 2 or 3). Other researchers have suggested a more profound alteration to the AES ShiftRows stage [49]. To elaborate, in every round, a value extracted from the round key will be used to determine the exact operation of the new ShiftRows operation. The technique of extracting this value is inspired by the DeoxyriboNucleic Acid (DNA) structure. It can be argued that the ShiftRows stage in the AES has been selected to achieve a specific diffusion property, that is, to ensure that every column bytes will be scattered among all other bytes. This property is essential in the proof of resistance to differential and linear attacks [48]. Hence, it cannot be granted whether this new technique has a real improvement to the cipher security or not. Moreover, there is no analysis for the cipher performance, in terms of time needed for encryption and decryption. Hence, due to the complexity of the modified ShiftRows stage, it may significantly reduce the time performance of the cipher.

Furthermore, the matrix used for column mixing can be modified. The RotWord operation in the key expansion can also be modified, etc. In [50], they increased the number of rounds to 16 and dramatically increased the key length to 320 bits, rather than 128, 192 or 256 bits (see figure 2-3).

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

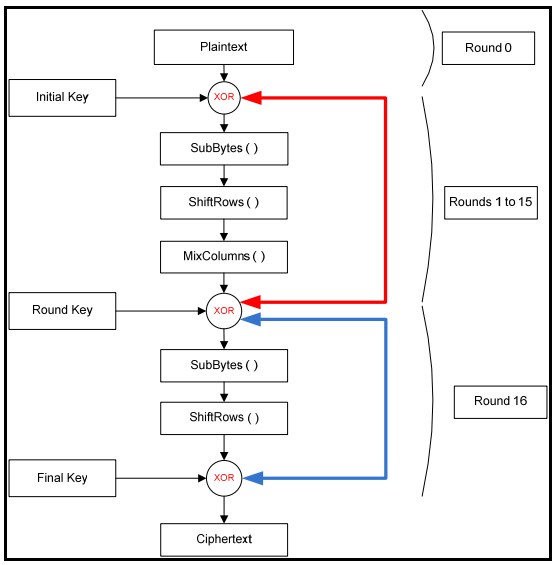

Figure 3 : Modified AES with 16 Rounds: Encryption Module [50]

Such tweaks can make cryptanalyzing the AES harder for intruders. However, it was proven in [51] that to retrieve these new parameters is relatively easy.

Another category of AES-like algorithms aims mainly to modify the internal operation of the algorithm for performance reasons. This includes [52] in which they replace the relatively CPU intensive operation of Column Mixing by another layer of customized SubBytes operation. Although the overall performance has been increased notably, the security has been degraded.

3.5. Chaotic Encryption

Some researchers have chosen to follow a relatively different path for designing a new approach for encryption, which is Chaotic Encryption. Since Edward Lorenz introduced Chaos Theory in the 1960s, it has attracted the interest of scholars from different disciplines including cryptographers [53]. Chaos systems are simply nonlinear systems that have several properties including:

1. Ergodicity "similar to the confusion" property of the symmetric ciphers.

2. Sensitivity to initial conditions (future states are very hard to predict given the initial states).

3. Topological Mixing "similar to multi-round transformation".

For these reasons, chaos systems can serve as a suitable encryption tool. Several encryption schemes that rely on chaotic systems have been introduced in the last few years [54-56].

Another example of the latter class of encryption techniques has been introduced in [57]. In this research, they proposed an algorithm for image encryption. At the start, they will generate 256 different S-Boxes with good algebraic properties (i.e. nonlinearity, avalanche criterion, bit independence...etc.). The plain image will be decomposed into 3 layers: Red, Green, and Blue. Pixels of each layer will be substituted using one of the S-Boxes that has been generated above. The identity of the S-Box that will be used to substitute each pixel will be determined according to a chaotic system. After encrypting all three layers, they will be composed together to form the encrypted image. Another similar approach has been introduced in [58]. In this approach, the encryption is performed using three chaotic maps and S8 Liu J S-boxes. The encryption operation includes 4 rounds, and each round contains cyclic shifting, row and column permutation and Substitution with the S-Boxes. According to the performance and security analysis provided in the paper, the approach has leverage on many widely used ciphers. Moreover, it has been mentioned that the introduced approach can recover the encrypted data even if they have been slightly corrupted by intentional or unintentional noise. It can be noted from [57] and [58] among other researches that S-Boxes are considered a vital component in the operation of each cipher. A good S-Box must meet several criteria which have been addressed thoroughly in the literature. To make things easier, in [59] they suggested an algorithm that can be used to generate S-Boxes with the required features.

Although chaotic encryption is one of the current active research areas, it has its limitations such as the requirement of floating-point computations, conversion of operations, infinite periodicity and expensive hardware implementation [41]. Moreover, chaotic encryption algorithms remain deterministic algorithms that do not change how they work. Accordingly, there is no dynamism or polymorphism in the operation of chaotic encryption algorithms.

3.6. Steganography

Steganography algorithms, techniques, and protocols can be considered as a variant of

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

encryption algorithms. In steganography, the existence of secret communication is hidden by hiding the secret data inside an apparently innocent carrier, instead of scrambling the data like encryption algorithms [60].

Steganography has ancient techniques, which includes:

1. Character marking: over-writing selected letters of printed text by pencil. The hidden traces are not normally appearing unless the paper is tilted.

2. Invisible ink.

3. Letters puncturing: applying small punctures on selected letters that are normally not appearing except when holding the paper in front of a light.

On the other hand, modern steganography uses computer files to achieve the same goal. Different file types can be used as a carrier and the steganography techniques usually depend on the file type in use. The most popular branch of steganography algorithms relies on using images as a carrier and replacing the Least Significant Bits (LSBs) of the whole or specific pixel in the image to hide the secret data (See figure 2-4) [61, 62]. Other file types can be used too, including text files, video files or even executable files.

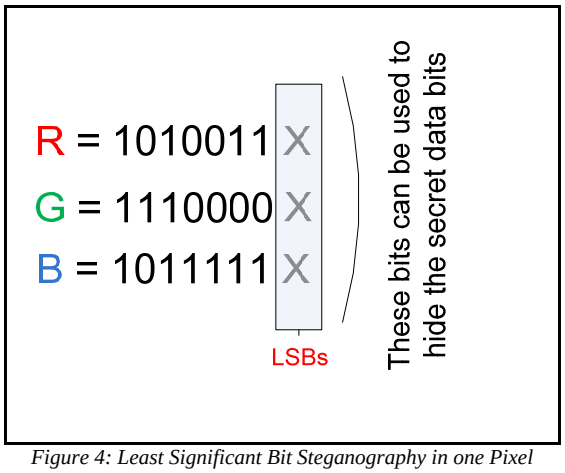

Figure 4: Least Significant Bit Steganography in one Pixel

The key problem of using steganography algorithms is that if the opponent becomes aware that a secret communication is taking place, he can threaten the whole model. Some solutions have been introduced, most of them rely on using a sort of encryption and a key to protect the data from steganalysis techniques [7].

On the other hand, the effect on the performance due to using encryption and steganography is usually high [7]. Moreover, the size of the cover file "carrier" will be added to the transmitted message. This will waste valuable network capacity. One more challenge is the carrier selection. For example, if you need to send a long message, you must select a huge file as a carrier, as most steganographic techniques use only a fraction of the carrier file to hide the real message. Furthermore, this carrier file should be changed with every new message or the opponent may suspect the communication. The file should contain relevant data that can be considered reasonable for the opponent. Otherwise, the opponent will question this message. To clarify, assume the opponent is monitoring the communications between one military personnel and his commander. If the carrier was a romantic novel (due to a poor selection of the carrier file in the sender side) the opponent will certainly suspect the message.

3.7. Security through Obscurity

This ubiquitous concept includes:

1. Renaming built-in system components (user accounts, operating system utilities…etc.).

2. Use non-standard network ports to provide services.

3. Install applications to non-default directories.

The proponents of obscurity techniques use the following arguments in support of using obscurity measurements to secure the relying system [63]:

1. Reduce the probability of successful attacks.

2. Slowing down the determined attacker.

On the other hand, the opponents of obscurity have the following arguments:

1. These techniques do not provide "real security", as the determined attacker will always be able to find these obscured configurations.

2. Such configurations might significantly affect the system interoperability.

3. Some obscurity arrangements might conflict with the automatic updates. As a result, the

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

system admin might ignore or postpone updating the system due to the complexity of re-setting the modified parameters. This exposes the system to real threats, as the system will not be up-to-date, thus vulnerable.

Therefore, it is of high importance to well consider the cost of using obscurity arrangements before applying them. Moreover, it is judicious to apply these arrangements at the top of other robust security safeguards (e.g. encryption) rather than relying on it as the only layer of protection.

One variant of security through obscurity is to build your encryption algorithm, without sharing its operation details with anyone except a small group of trusted people who need to engage in confidential communication. This approach is generally not encouraged because it eliminates interoperability. Moreover, to decide whether a cipher is secure or not, it must be made available for a wide range of experts. These experts will state whether the cipher is secure or not after applying several tests. If the corporation decided to use their own cipher without enough wide-scale testing, they might have a false feeling of confidence toward their cipher robustness. As mentioned before, Kerckhoffs has introduced this concept over more than a century [64, 65]. Since then, it has got a great consensus among security experts and cryptographers.

3.8. Encryption on Demand

In this model [66], they developed a web-based application that is used to encrypt the secret messages. The two parties' who wish to engage in is the secret communication agree on a number in the range 1 to 999,999,999. This number is called the Tailoring ID (TID). The method of agreeing on the TID (i.e. sharing the TID) is considered out of the scope of their research. This TID is sent to the server, which "cooks" a specific software package that includes a specific encryption algorithm with its parameters and key. The algorithm is either AES, DES, RSA or ECC. The encryption algorithm, key, and key length are assigned by the algorithm according to the TID value. This package is then sent back only to the clients who provided this TID. Clients can then use this package to encrypt their work by writing or pasting the secret message to this software. The package will calculate the equivalent ciphertext of the entered text, and the user can copy the encrypted version of the message and send it to the other party. All the cryptographic parameters will be hidden from everyone, including the communicating parties. Only the server knows which cipher, key…etc. has been assigned to this TID package. In [67] the same concept has been slightly optimized to fit multimedia data (See figure 2-5).

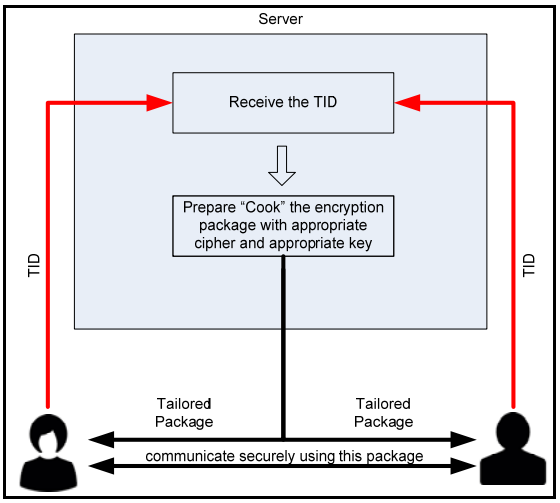

Figure 5: Encryption on Demand Explained (Amato et al., 2014)

This model has the following limitations:

1. Encryption operations should be transparent for the user. Enforcing the user to explicitly visit a webpage, and download the package is a tedious task. If this is the only solution for encryption inside specific corporate, many users will just ignore encrypting their messages, which will significantly decrease the overall level of security inside the corporation.

2. As mentioned explicitly inside their research paper [66], The opponent can intercept the client, and got a copy of the encryption package. Therefore, he can now decrypt all secret messages exchanged between communicating parties.

3. As the TID ranges from one to 999,999,999, the opponent needs to guess the TID from this range (i.e. 999,999,999 different options) rather than trying all the different combinations for the AES key. The shortest AES key is 128 bits length, which is 340,282,366,920,938,463,463,374,607,431,768,211,456 different options.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.9. A hybrid symmetric and asymmetric approach

Probably, the most intuitive example for symmetric and asymmetric ciphers' hybrids is the Secure Sockets Layer (SSL) protocol. Using this well-known protocol, the client and the server exchange all the needed material to establish a secure session. To elaborate, the SSL protocol is composed of four sub-protocols. One of these sub-protocols is the SSL Handshake Protocol. During the SSL Handshake Protocol, the client typically generates a pseudorandom value known as pre-master-secret. This pre-master-secret is typically encrypted using the RSA using the server public key, after verifying and validating the server certificate. The pre-master-secret is used to generate a symmetric encryption key that will be used with a symmetric cipher to encrypt the messages exchanged in this session.

It worth mentioning that one phase in the SSL Handshake Protocol involves negotiating the symmetric cipher that will be used for encrypting the payloads. In this phase, the client will send a list of ciphers ordered according to his preference. The server will pick one cipher taking into account the order of preference provided by the client. This cipher will typically be used to secure all the messages in a given session [8]. Although, the context is different, the reader can note that it is normal for the client to support several ciphers, and to pick one of these ciphers according to some considerations for the purpose of encrypting the payload.

Apart from the SSL, there are some efforts to design other symmetric and asymmetric ciphers' hybrids. For instance, in this work [68], they are using the conventional integration between RSA and AES to resolve the key sharing obstacle and utilize the high performance of the symmetric encryption algorithms. The idea is to generate a random AES key to be used for encrypting the secret data. After encrypting the data, the RSA will be used to encrypt the symmetric key. Now the encrypted data can be shared, and the private key holder (i.e. legitimate receiver) can decrypt the AES key (used for encrypting the message payload) and, consequently, decrypt the payload.

The only worth mentioning contribution of this work is that the authors enhanced the process of generating the public-private keys pair by introducing a novel approach that finds prime numbers faster. Therefore, this variant of the RSA that has been proposed is more efficient compared to the standard RSA. Consequently, the whole model will be more efficient. On the other hand, the whole model is basically a trivial version of the SSL Protocol. Moreover, even the real contribution of the paper (i.e. enhancing the RSA performance by expediting the process of prime numbers generation) comes at the cost of the level of assurance in the primality of the selected parameters. The reader is reminded that the RSA relies solely on the difficulty of factorizing prime numbers. Therefore, using numbers with low primality can pose the RSA algorithm to potential attacks.

4. SPECIAL PURPOSE ALGORITHMS

Some algorithms have been proposed to meet the requirements of a specific environment. These are not general-purpose algorithms, and usually, they will not fit except the targeted environment. Next paragraphs will make a brief overview of some of these algorithms.

4.1.1. Searchable Symmetric Encryption

This is a direction of research in which the encrypted data, typically a database, is encrypted yet the client can search for keywords in the encrypted database. The need for this category of algorithms arose as a requirement to protect the very sensitive data in semi-trusted servers that provide storage service. Several attempts that satisfy this requirement has been discussed in the literature, including [69] and [70].

Although these algorithms provide a symmetric encryption function, their main design goal is to facilitate the requirement of searching within encrypted data. Therefore, it can be classified as special-purpose encryption algorithms. This study focuses mainly on a generic encryption algorithm that provides a high level of security and acceptable performance.

4.1.2. Polymorphic Encryption Key

In [71], they introduced a model for broadcast traffic encryption. A piece of the data that will be used to calculate the key is built-in the receiving side machine. When the encrypted traffic has been received, a specific part of that traffic plus the built-in key will be passed together to a cryptographic function. The output of this function will be the decryption key to be used for decrypting

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

the traffic. The same concept has been developed and optimized in [72] and [73].

This model is designed solely to protect broadcast traffic. Accordingly, its scope is limited only to that context.

4.1.3. Securing the TFTP protocol

According to [74], the Trivial File Transfer Protocol (TFTP) is a widely known protocol that is used mainly for transfer, configure, or update low storage devices including Wireless Access Points (WAPs). This protocol does not provide security services, thus the authors enhanced the protocol by providing the appropriate security mechanisms. Briefly speaking, the following steps are applied to every new communication:

1. A Diffie Hellman key exchange algorithm (DHKE) is applied to securely agree on a session key. In DHKE the communicating parties exchange two different values that help in calculating the key. This process might fall to a Man-In-the-Middle attack (MITM), as no previous authentication took place, thus the attacker can impersonate the identity of Alice to Bob or vice versa. To avoid this, the value that should be sent from the client to the server in the DHKE will be built-in the server. Therefore, they will be no enough information for the opponent to construct the session key.

2. A compression function will be applied using the Huffman coding algorithm. This phase has been applied to reduce the size of the data packets to be sent.

3. The data will be encrypted using the AES algorithm.

4. The hash code for the data will be calculated using the SHA-2 algorithm, and sent separately, so the recipient can regenerate the digest in his side, and compare it to the received digest.

The authors claimed that the overall performance would be better than the performance of the default protocol. The reason for this improvement according to the authors is due to the compression process that decreases the packets size, thus reduce the time consumed in processing the file.

The originality of this research relies in applying these well-known, widely used security measures to secure the TFTP. An additional contribution is the slight modification applied to the DHKE protocol to resolve the authentication difficulty, which exposes the protocol to the MITM attack.

On the other hand, it can be argued that all the used mechanisms – other than stated above – is a conventional mechanism used to secure a wide range of applications.

5. CONCLUSION

Many researchers have attempted to develop new protocols to utilize a set of other algorithms or to modify the design of encryption algorithms. However, most of these models either have a significantly poor performance or relies solely on the privacy of the encryption keys, without attempting to hide the identity or the operation details of the used encryption algorithm. It can be concluded that the current encryption approaches give the opponent a significant advantage, as he will be aware of which algorithm was used to encrypt the message. As a result, if the opponent has managed to obtain a technique to cryptanalyze the encryption algorithm, the security of the message will be threatened. One exception is the research work that supports building your own cipher without sharing it with anyone outside a specific trusted group of individuals. As mentioned previously this approach generally unacceptable [64, 75]. Another exception is to use dynamic encryption. However, current dynamic ciphers have a considerably poor performance or does not have provable security.

Therefore, there is probably a need to introduce a dynamic cipher that can operate unpredictably. This cipher must also provide interoperability, provable security, and high performance. Furthermore, the implementation cost of this cipher must be tolerable.

REFRENCES:

[1] M. Cardona, T. Kretschmer, and T. Strobel, "ICT and productivity: conclusions from the empirical literature," *Information Economics and Policy,* vol. 25, no. 3, pp. 109-125, 2013.

[2] P. Hunton, "The growing phenomenon of crime and the internet: A cybercrime execution and analysis model," *Computer Law & Security Review,* vol. 25, no. 6, pp. 528-535, 2009.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[3] R. Anderson *et al.*, "Measuring the cost of cybercrime," in *The economics of information security and privacy*: Springer, 2013, pp. 265-300.

[4] W. J. Caelli, "Trusted… or… trustworthy: the search for a new paradigm for computer and network security," *Computers & Security,* vol. 21, no. 5, pp. 413-420, 2002.

[5] S. Singha and M. Sen, "Encoding algorithm using bit level encryption and decryption technique," in *Computer, Electrical & Communication Engineering (ICCECE), 2016 International Conference on*, 2016, pp. 1-4: IEEE.

[6] S. Mocanu, A. Duluta, D. Merezeanu, and R. Pietraru, "Improved Security Based on Combined Encryption and Steganography Techniques," *Studies in Informatics and Control,* vol. 26, no. 1, pp. 115-126, 2017.

[7] A. Altigani and B. Barry, "A hybrid approach to secure transmitted messages using advanced encryption standard (AES) and Word Shift Coding Protocol," in *Computing, Electrical and Electronics Engineering (ICCEEE), 2013 International Conference on*, 2013, pp. 134-139: IEEE.

[8] W. Stallings, *Cryptography and network security: principles and practice*. Pearson, 2016.

[9] J. A. Carlson, "Method for secure communication using asymmetric and symmetric encryption over insecure communications," ed: Google Patents, 2019.

[10] M. Baldi, F. Chiaraluce, L. Incipini, and M. Ruffini, "Code-based physical layer secret key generation in passive optical networks," *Ad Hoc Networks,* 2019.

[11] C. Paar and J. Pelzl, *Understanding cryptography: a textbook for students and practitioners*. Springer Science & Business Media, 2009.

[12] A. Zaidan, B. Zaidan, M. Abdulrazzaq, R. Raji, and S. Mohammed, "Implementation stage for high securing cover-file of hidden data using computation between cryptography and steganography," *International Association of Computer Science and Information Technology (IACSIT),* vol. 19, pp. 482-489, 2009.

[13] A. Naji, A. Zaidan, B. Zaidan, and I. A. Muhamadi, "Novel approach for cover file of hidden data in the unused area two within EXE file using distortion techniques and advance encryption standard," *Proceeding of World Academy of Science Engineering and Technology (WASET),* vol. 56, no. 5, pp. 498-502, 2010.

[14] M. Abomhara, O. Zakaria, O. O. Khalifa, A. Zaidan, and B. Zaidan, "Enhancing Selective Encryption for H. 264/AVCUsing Advanced Encryption Standard," *International Journal of Computer and Electrical Engineering,* vol. 2, no. 2, p. 223, 2010.

[15] J. Longo, D. P. Martin, L. Mather, E. Oswald, B. Sach, and M. Stam, "How low can you go? Using side-channel data to enhance brute-force key recovery," *IACR Cryptology ePrint Archive,* vol. 2016, p. 609, 2016.

[16] F.-H. Hsiao and P.-H. Lin, "Applying Triple Data Encryption Algorithm to a Chaotic Systems: TS Fuzzy Model-Based Approach," in *International Conference on Information Science and Applications*, 2018, pp. 53-66: Springer.

[17] M. Aljohani, I. Ahmad, M. Basheri, and M. O. Alassafi, "Performance Analysis of Cryptographic Pseudorandom Number Generators," *IEEE Access,* vol. 7, pp. 39794-39805, 2019.

[18] V. Kapoor and S. Dey, "Symmetric Algorithms II," *Emerging Security Algorithms and Techniques,* p. 97, 2019.

[19] E. Biham, R. Anderson, and L. Knudsen, "Serpent: A new block cipher proposal," in *International Workshop on Fast Software Encryption*, 1998, pp. 222-238: Springer.

[20] R. L. Rivest, M. J. Robshaw, and Y. L. Yin, "RC6 as the AES," in *AES Candidate Conference*, 2000, pp. 337-342.

[21] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson, "Twofish: A 128-bit block cipher," *NIST AES Proposal,* vol. 15, no. 1, pp. 23-91, 1998.

[22] D. Coppersmith, "The Data Encryption Standard (DES) and its strength against attacks," *IBM journal of research and development,* vol. 38, no. 3, pp. 243-250, 1994.

[23] C. Burwick *et al.*, "MARS-a candidate cipher for AES," *NIST AES Proposal,* vol. 268, p. 80, 1998.

[24] C. K. B. Röllgen, "Block cipher," ed: Google Patents, 2010.

[25] A. Biryukov and D. Khovratovich, "Related-key cryptanalysis of the full AES-192 and AES-256," *Advances in*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Cryptology–ASIACRYPT 2009, pp. 1-18, 2009.

[26] A. Bogdanov, D. Khovratovich, and C. Rechberger, "Biclique cryptanalysis of the full AES," *Advances in cryptology–ASIACRYPT 2011,* pp. 344-371, 2011.

[27] A. Biryukov, O. Dunkelman, N. Keller, D. Khovratovich, and A. Shamir, "Key recovery attacks of practical complexity on AES-256 variants with up to 10 rounds," in *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, 2010, pp. 299-319: Springer.

[28] O. Dunkelman, N. Keller, and A. Shamir, "Improved single-key attacks on 8-round AES-192 and AES-256," *Journal of Cryptology,* vol. 28, no. 3, pp. 397-422, 2015.

[29] K. Zhao, J. Cui, and Z. Xie, "Algebraic Cryptanalysis Scheme of AES-256 Using Gröbner Basis," *Journal of Electrical and Computer Engineering,* vol. 2017, 2017.

[30] B. Tao and H. Wu, "Improving the biclique cryptanalysis of AES," in *Australasian Conference on Information Security and Privacy*, 2015, pp. 39-56: Springer.

[31] L. Grassi, C. Rechberger, and S. Rønjom, "Subspace trail cryptanalysis and its applications to AES," *IACR Transactions on Symmetric Cryptology,* vol. 2016, no. 2, pp. 192-225, 2017.

[32] V. Karuvandan, S. Chellamuthu, and S. Periyasamy, "Cryptanalysis of AES-128 and AES-256 block ciphers using lorenz information measure," *Int. Arab J. Inf. Technol.,* vol. 13, no. 6B, pp. 1054-1060, 2016.

[33] S. Kakarla, S. Mandava, D. Saha, and D. R. Chowdhury, "On the practical implementation of impossible differential cryptanalysis on reduced-round AES," in *International Conference on Applications and Techniques in Information Security*, 2017, pp. 58-72: Springer.

[34] R. Li and C. Jin, "Meet-in-the-middle attacks on 10-round AES-256," *Designs, Codes and Cryptography,* vol. 80, no. 3, pp. 459-471, 2016.

[35] C. E. Shannon, "Communication theory of secrecy systems," *Bell Labs Technical Journal,* vol. 28, no. 4, pp. 656-715, 1949.

[36] О. Ляшук, "Highly efficient method of data protection based on multilayer hybrid encryption," *BULLETIN of National Technical University of Ukraine. Series RADIOTECHNIQUE. RADIOAPPARATUS BUILDING,* no. 56, pp. 144-151, 2014.

[37] B. Minaud and Y. Seurin, "The iterated random permutation problem with applications to cascade encryption," in *Annual Cryptology Conference*, 2015, pp. 351-367: Springer.

[38] C. B. Roellgen, "The Polymorphic Medley Cipher Version 2: 128 bit block length,128 .. 1024 bit key length," White Paper 2013.

[39] I. A. Shoukat, "Muli-operation Data Encryption Mechanism Using Dynamic Data Blocking and Randomized Substitution," Universiti Teknologi Malaysia, 2016.

[40] H. Tang, Q. T. Sun, X. Yang, and K. Long, "A network coding and DES based dynamic encryption scheme for moving target defense," *IEEE Access,* vol. 6, pp. 26059-26068, 2018.

[41] H. N. Noura, A. Chehab, and R. Couturier, "Efficient & secure cipher scheme with dynamic key-dependent mode of operation," *Signal Processing: Image Communication,* vol. 78, pp. 448-464, 2019.

[42] C.-P. Young, C.-C. Chia, L.-B. Chen, and J. Huang, "NCPA: A Scheduling Algorithm for Multi-cipher and Multi-mode Reconfigurable Cryptosystem," in *Intelligent Information Hiding and Multimedia Signal Processing, 2008. IIHMSP'08 International Conference on*, 2008, pp. 1356-1359: IEEE.

[43] H. Singh and P. Singh, "Enhancing AES using Novel Block Key Generation Algorithm and Key Dependent S-boxes," *Cyber-Security and Digital Forensics,* p. 30, 2016.

[44] Ü. Çavuşoğlu, S. Kaçar, A. Zengin, and I. Pehlivan, "A novel hybrid encryption algorithm based on chaos and S-AES algorithm," *Nonlinear Dynamics,* vol. 92, no. 4, pp. 1745-1759, 2018.

[45] R. B. Antonio, A. M. Sison, and R. P. Medina, "Performance Analysis of the Modified Generated S-Box for Advanced Encryption Standards," in *Proceedings of the 2019 2nd International Conference on Data Science and Information Technology*, 2019, pp. 117-121: ACM.

[46] A. Seghier and J. Li, "AES Based on Key Dependently Nonlinear Redundant S-Box," in *ICC 2019-2019 IEEE International*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Conference on Communications (ICC), 2019, pp. 1-6: IEEE.

[47] A. Seghier, J. Li, and D. Z. Sun, "Advanced encryption standard based on key dependent S-Box cube," *IET Information Security,* 2019.

[48] J. Daemen and V. Rijmen, "AES proposal: Rijndael," 1999.

[49] A. H. Al-Wattar, R. Mahmod, Z. A. Zukarnain, and N. I. Udzir, "A New DNA-Based Approach of Generating Key-dependent ShiftRows Transformation," *arXiv preprint arXiv:1502.03544,* 2015.

[50] P. Kumar and S. B. Rana, "Development of modified AES algorithm for data security," *Optik-International Journal for Light and Electron Optics,* vol. 127, no. 4, pp. 2341-2345, 2016.

[51] C. Clavier, Q. Isorez, D. Marion, and A. Wurcker, "Complete reverse-engineering of AES-like block ciphers by SCARE and FIRE attacks," *Cryptography and Communications,* vol. 7, no. 1, pp. 121-162, 2015.

[52] F. V. Wenceslao Jr, "Performance efficiency of modified AES algorithm using multiple S-Boxes," *International Journal of New Computer Architectures and their Applications (IJNCAA),* vol. 5, no. 1, pp. 1-9, 2015.

[53] K. Bora and M. A. Wadood, "Implementation and Analysis of Symmetric Key Encryption Technique Using Chaos Theory," *European Journal of Advances in Engineering and Technology,* vol. 2, no. 1, pp. 86-90, 2015.

[54] X. Yang, X. Hu, Z. Shen, H. He, W. Hu, and C. Bai, "Physical layer signal encryption using digital chaos in OFDM-PON," in *2015 10th International Conference on Information, Communications and Signal Processing (ICICS)*, 2015, pp. 1-6: IEEE.

[55] Z. Wang and S. Chen, "A Chaos-Based Encryption Scheme for DCT Precoded OFDM-Based Visible Light Communication Systems," *Journal of Electrical and Computer Engineering,* vol. 2016, 2016.

[56] X. Chai, Y. Chen, and L. Broyde, "A novel chaos-based image encryption algorithm using DNA sequence operations," *Optics and Lasers in engineering,* vol. 88, pp. 197-213, 2017.

[57] I. Hussain, A. Anees, A. H. AlKhaldi, A. Algarni, and M. Aslam, "Construction of chaotic quantum magnets and matrix Lorenz systems S-boxes and their applications," *Chinese Journal of Physics,* 2018.

[58] I. Hussain, A. Anees, M. Aslam, R. Ahmed, and N. Siddiqui, "A noise resistant symmetric key cryptosystem based on S 8 S-boxes and chaotic maps," *The European Physical Journal Plus,* vol. 133, pp. 1-23, 2018.

[59] A. Altaleb, M. S. Saeed, I. Hussain, and M. Aslam, "An algorithm for the construction of substitution box for block ciphers based on projective general linear group," *AIP Advances,* vol. 7, no. 3, p. 035116, 2017.

[60] B. M. Kumar and Y. Thakur, "An Introduction to Steganography Techniques in the Field of Digital Image Processing," *International Journal of Engineering Science,* vol. 13495, 2017.

[61] W. William, A. Osofisan, and M. Asanbe, "A Lookup XOR Cryptography for High Capacity Least Significant Bit Steganography," ed: IJAIS, 2016.

[62] S. N. Gowda and S. Sulakhe, "Block Based Least Significant Bit Algorithm For Image Steganography," 2016: Annual Int'l Conference on Intelligent Computing, Computer Science & Information Systems (ICCSIS-16).

[63] D. Stuttard, "Security & obscurity," *Network Security,* vol. 2005, no. 7, pp. 10-12, 2005.

[64] B. Norouzi and S. Mirzakuchaki, "Breaking a novel image encryption scheme based on an improper fractional order chaotic system," *Multimedia Tools and Applications,* vol. 76, no. 2, pp. 1817-1826, 2017.

[65] A. Kerckhoffs, *La cryptographie militaire, ou, Des chiffres usités en temps de guerre: avec un nouveau procédé de déchiffrement applicable aux systèmes à double clef.* Librairie militaire de L. Baudoin, 1883.

[66] G. Samid, "Encryption-On-Demand: Practical and Theoretical Considerations," *IACR Cryptology ePrint Archive,* vol. 2008, p. 222, 2008.

[67] D. Amato, P. Juan, and M. J. Vénere, "Encrypting video and image streams using OpenCL code on-demand," *CLEI Electronic Journal,* vol. 17, no. 1, pp. 6-6, 2014.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[68] C. Liang, N. Ye, R. Malekian, and R. Wang, "The hybrid encryption algorithm of lightweight data in cloud storage," in *Agent, Multi-Agent Systems and Robotics (ISAMSR), 2016 2nd International Symposium on*, 2016, pp. 160-166: IEEE.

[69] G. Asharov, M. Naor, G. Segev, and I. Shahaf, "Searchable symmetric encryption: Optimal locality in linear space via two-dimensional balanced allocations," in *Proceedings of the 48th Annual ACM SIGACT Symposium on Theory of Computing*, 2016, pp. 1101-1114: ACM.

[70] I. Demertzis and C. Papamanthou, "Fast Searchable Encryption With Tunable Locality," in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 1053-1067: ACM.

[71] J. B. Geagan and D. B. Ponceleon, "Polymorphic encryption key allocation scheme," ed: Google Patents, 2016.

[72] D. B. Ponceleon, "Polymorphic encryption key matrices," ed: Google Patents, 2016.

[73] J. B. Geagan III and D. B. Ponceleon, "Polymorphic encryption key allocation scheme," ed: Google Patents, 2017.

[74] N. Mohamed, Y. Yussoff, M. Isa, and H. Hashim, "Symmetric encryption using pre-shared public parameters for a secure TFTP protocol," *Journal of Engineering Science and Technology,* vol. 12, no. 1, pp. 098-112, 2017.

[75] G. Blakley, "Twenty years of cryptography in the open literature," in *Security and Privacy, 1999. Proceedings of the 1999 IEEE Symposium on*, 1999, pp. 106-107: IEEE.