Implementation of Security Enhancement in AES by Inducting Dynamicity in AES S-Box

Gousia Nissar, Dinesh Kumar Garg, Burhan Ul Islam Khan

Abstract*: Advance Encryption Standard (AES) supersedes Data Encryption Standard (DES) and is the best known and most widely used block cipher. As for now, there are no known practical attacks that would allow anyone to read correctly implemented AES encrypted data. However, several theoretical attacks have been announced until now. A theoretical attack called Biclique Attack is known to have broken Full AES and requires $2^{126.1}$, $2^{189.7}$, $2^{254.4}$ operations to recover an AES-128, AES-192, AES-256 respectively. Biclique Attack is faster than Brute force attack by a factor of four. As such, these theoretical attacks are of high computational complexity; they do not threaten the practical use of AES in any way. However, attacks always get better; they never get worse. As the technology evolves, successful attacks (using Quantum Computing and faster GPU) against AES may turn up, and they may be difficult to ignore. In this study, we aim to enhance the security prospects of AES with the inclusion of Dynamicity character in AES S-Box for increased resilience against Brute Force Attack and Biclique Attack, and hashing technique is combined with AES algorithm to achieve variance in security using MD4, SHA3 or SHA5. The novel key dispersion technique is introduced to increase the avalanche effect of AES algorithm.

***Keywords*: AES, DES, Encryption, S-Box.**

I. INTRODUCTION

The quality and exploration of shielding data from unwanted people by changing it over into a non-conspicuous form from the attackers while transmitting and storing the data is called cryptography. Information cryptography principally is the scrambling of the substance of information, for example, content, sound, picture, video et cetera to make the information scrambled, undetectable or distorted amid transmission or capacity called Encryption. The reverse of information encryption is information Decryption. Data Cryptography includes three distinct instruments: Symmetric-key Encryption, Asymmetric key encryption, and Hashing [1]. Symmetric-key encryption [2] uses a single secret key for both encryption and decoding. In Asymmetric Key encryption, there are two keys: Public key utilized for encryption and private key used for decryption [3]. Symmetric figures are partitioned into two classes: stream Cipher and block cipher. In a stream figure, encryption and decryption are done one bit/byte at once while as in stream cipher, a group of bits/bytes are encoded and decoded together. A number of cryptographic algorithms have been suggested, for example, the Data Encryption Standard (DES), the Advanced Encryption Standard (AES), the Elliptic Curve Cryptography (ECC), and different other algorithms. Cryptographic algorithms offer security to the web by preserving the integrity of data [4]. Bunches of scientists and programmers are continually endeavouring and trying to find weaknesses to break these algorithms by utilizing most grounded side channel and brute force assaults [5][6]. Some assaults were triumphant as it was the situation for the Data Encryption Standard in 1993, where the distributed cryptanalysis assault could break the DES. As we are aware, the security quality of Data Encryption Standard (DES) was not good enough to adjust the new needs. In October of 2000, the Rijndael algorithm was chosen by the National Institute of Standards and Technology (NIST) as the advanced encryption standard (AES), which was created by Joan Daemen and Vincent Rijmen, to supplant the DES. Currently, Rijndael is the most well-known and generally utilized symmetric cryptosystem to help mass information encryption. It offers a decent “mix of suppleness, proficiency and security”. The discussion in this section concentrates on cryptography and the various cryptographic techniques.

Encryption technique where the sender and receiver both share a single secret key is referred to as Symmetric Encryption. Until June 1976, this is the only type of encryption known publicly says Diffie and Hellman (1976). In late 1970, prior to asymmetric key encryption, this is only type used [7]. Symmetric key ciphers are categorized/implemented into two: Block Ciphers and Stream Ciphers. The Stream Ciphers encrypt the plain text one bit at a time while as Block Ciphers work on plain text, but a given size of bits in it. The given size of bits here is termed as a block. Block Ciphers of existent generation have block sizes of 128 bits, i.e. 16 bytes. IBM’s Lucifer Cipher in the early 1970s has designed the Data Encryption Standard, and it was the first standard encryption model issued in July 1977 as the Federal Information Processing Standard [8]. DES encrypts 64-bit size of the input to a 64-bit output block size using a key of 56-bit size to customize the conversion, to carry on the process of decryption with the same key utilized for encryption. The absolute key size is 64 bits, and every 8th bit of the key, i.e. 8, 16, 24, 32, 40, 48, 56 and 64 are eliminated from 64 bits with only 56-bit key left. Here, 8 bits are exclusively used for parity checking and hence neglected.

DES pertains to the group of Feistel networks that are Symmetric Block Ciphers.

Manuscript published on 30 September 2019.
*****Correspondence Author(s)

Gousia Nissar, Department of CSE, SSCET Badhani (Pathankot) Gurdaspur, Punjab, India
Dinesh Kumar Garg, Department of CSE, SSCET Badhani (Pathankot) Gurdaspur, Punjab, India.
Burhan Ul Islam Khan*, Department of ECE, Kulliyyah of Engineering, UIAM Gombak, Malaysia.

Conventionally, Feistel networks are built in such a way that the encrypted text is divided into two halves. With the aid of key, a function is applied to the first half, the result is obtained, and the other half are operated with Boolean exclusive-or (XOR), and now the two halves are swapped [9]. Schneier and Kelsey (1996) have proposed a generalisation of this structure and most recent ciphers follow this structure. Structural similarity of encryption and decryption makes the Feistel structure suitable in ciphers. Denning (1997) says that DES was observed to have been implemented all over the world in 1996, with 44% of 1393 identified encryption products. DES in early 1998 was broken in 56 hours. Advancements in hardware and mathematical cryptanalysis make DES insecure and is thus used only in legacy systems since 1999. A competition was held in 1998 by NIST to the cryptographic community to replace DES [8]. AES - Advanced Encryption standard developed by Vincent Rijmen and Joan Daemen stood first and is also referred to as Rijndael.

At present, DES is thought of as being unsafe for most of the purposes, the main reason being the size of the key, i.e. 56-bit being too small. But the algorithm in the form of Triple DES is supposed to be secure enough, in spite of its theoretical attacks and with AES replacing DES, NIST has withdrawn DES as a standard. In 1998, NIST conducted a competition for the cryptography community to replace the defunct DES and AES - Advanced Encryption Standard stood as the winner being very suitable one in the competition where fifteen competing designs were presented and evaluated [8]. A modified form of Rijndael is AES in which the size of the block is limited to 128 bits. It has replaced DES, and in 2001 it is approved by US NIST [10] and is used worldwide currently. Feistel networks that are used in DES are not utilized in AES; it's based on substitution-permutation network and in which the size of the block is fixed to be 128-bits, with the size of the keys of 128, 192 and 256 bits [11]. Repetition of transformation rounds translates the input into output, i.e. the cipher text, these repetitions are specified by the size of the key used in the algorithm. The cycle counts for repetition are:

- For 128-bit keys, 10 cycles of repetition.
- For 192-bit keys, 12 cycles of repetition.
- For 256-bit keys, 14 cycles of repetition.

Every round of transformation performs several steps, along with the encryption key itself. The decryption is done by many reverse rounds utilizing the same key used for encryption (FIPS197, 2001).

This paper is comprised of various sections. Section-II discusses the problem statement followed by the research objectives in Section-III. Section-IV discusses the variety of schemes employed to enhance the security prospectus of AES. Section-V explain the system design. Section-VI illustrates the framework design of the proposed system and design implementation in Section-VII. Section-VIII provides the analysis of results, and finally, the concluding remarks have been discussed in Section-IX.

II. PROBLEM STATEMENT

Advanced Encryption Standard (AES) is utilized popularly over the world as the most secure algorithm for encryption available today. It has been utilized by the government of the U.S. to secure their private data. It is being implemented across the globe for encryption of classified information. For AES-192, there are attacks on seven [12, 13, 14], eight [13] and the full 12 rounds [15]. For AES-256, there are attacks on seven [16, 12], eight [13, 14] and the full 14 rounds [15]. This is due to the slower diffusion property in the key schedules for these variants compared to the AES-128. For the 12-round of AES-192, related-key type attacks manage to penetrate seven [17], eight [17, 18], nine [19] and ten [20] rounds. Similarly, for the 14-round of AES-256, related key attacks can be launched on nine [14, 13, 21] and ten [13, 21] rounds and are the best existing attacks on AES-192 and AES-256 (in the related-key attack model). The biclique [15], which is based on the meet-in-the-middle attack, is currently the best attack on all key size variants of the AES. Biclique Attack is known to have broken Full AES & requires 2126.1, 2189.7, and 2254.4 operations to recover an AES-128, AES-192, and AES-256 respectively. Biclique Attack is four times quicker than the Brute force attack. As such, these theoretical attacks are of computationally more complex, and hence do not intimidate the use of AES in any practical way.

Nevertheless, with time, attacks always improve. As the technology evolves, successful attacks (using Quantum Computing & Faster GPU) against AES may show up & it may not be possible to ignore them hence improvement in security in AES is needed to protect the sensitive data against the potential attacks. With static S- boxes fixing on the key, same input will be mapped to fixed output infinitely. This makes the underlying encryption systems prone to passive attacks traffic monitoring/analysis.

III. RESEARCH OBJECTIVES

1. To study various approaches and techniques that have been introduced in the prior literature for Cryptography particularly, Advanced Encryption Standard and deducing open issues from the same.
2. To modify the existing AES Algorithm
   a. To enhance the resilience of AES against various attacks, mainly Brute Force Attack & Biclique Attack.
   b. To Increase the resistance against Traffic analysis and Monitoring.
3. To improve the Confusion and diffusion factors in AES algorithm and hence increase the Avalanche Effect of AES Algorithm.
4. To do the mathematical analysis of the proposed algorithm.

IV. LITERATURE REVIEW

In the year 1997, the United States National Institute of Standards and Technology (NIST, formerly NBS) asked the public, including academics and professionals, to submit new cryptography algorithms as possible candidates to become the New Advanced Encryption Standard (AES) in order to substitute the outdated DES algorithm for use with sensitive, but not confidential government data.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

There were five leading candidates for the new standard, namely, Rijndael, Serpent, 2fish, RC6 and MARS. In November 2001, Rijndael was selected to be the new AES standard. It proved to be the most efficient and the fastest among all the contending algorithms. It is likewise executed on a wide variety of platforms and is extendable to various lengths of blocks and keys [22].

A comprehensive survey of different issues relating to security and validation of accessing private and exceptionally sensitive data that have been considered by numerous scientists. Research work led features different strategies embraced in the past to alleviate different sorts of assaults on authentication arrangement of the client and takes care of issues related to security.

A. S-Box based Schemes

For development in security, the static S-Box, which is used in the substitution step of transformation in AES, is changed. The recommended S-Box is arbitrary and depends upon the idea put forward by [23], of Key-Dependent S-Boxes and furthermore by the algorithm proposed by Knuth. The proposed S-Boxes because of the strong properties utilize some portions of original parts of the AES.

- Dynamic S-Box

The dynamic S-Box put forward as recommended in the proposition in [24] is characterized as under:

1. According to the unique AES algorithm, $Nr-1$ round keys are driven from the cipher key.
2. In round j where $1 <= j < Nr$ a random string of 128 bits is created by utilizing the K_j as input to an arbitrary number generator (hash function).
3. This 128-bit string is then divided into 16-word, each word is of 8-bit length.
4. A state which consists of 16-word is what AES works on. In round j there are:
   a. A 16-word long random string which characterizes the state.
   b. A 16-word long random string.
5. If the word to be replaced is w_{ji}, the equivalent random word being R_{ji}, the unique AES S-Box being S, the unique AES inverse S-Box being IS, and the two are represented as one-dimensional arrays. The proposed substitution step progress toward becoming:

- Random S-Box

The Proposed plan for the Random S-Box is characterized by [7] and is as per the following:

6. $Nr-1$ round keys are driven from the cipher key.
7. In the round j where $1 <= j < Nr$, a random string of 256 bits length is created by utilizing the K_j as input to an arbitrary number generator (hash function).
8. The Random S-Box is created by utilizing the random string, which was produced in the past, as indicated by the algorithm proposed by Knuth.

Strengths:

- The results of the proposed systems have provided a solution for achieving dynamicity in s-boxes and hence increases the security for AES.
- The results of the conducted tests on the proposed design are extremely encouraging. Actually, the aftereffects of AES utilizing dynamic S-Box were near the consequences of actual AES itself. Then again, the aftereffects of AES utilizing the random S-Box outflank those of the actual AES algorithm.

Limitations:

- The proposed design does not counter biclique attack, which depends on the meeting-the-middle assault and is right now the most well-known assault on all key size variations of the AES.
- NIST tests which were conducted on the new attuned AES take into consideration the 128-piece key length. Thus, the proposed configuration should be tried by the rest of the NIST tests which treat different lengths of the key.
- The runtime complexity is very high in both the proposed designs.
- No impact on traffic analysis and monitoring.
- No improvement in resistance against Brute force attack.

B. Key Scheduling based Schemes

There are numerous cryptanalytic assaults against the AES. It can be noticed that the key related attacks are especially compelling in assaulting AES-192 and AES-256. This is primarily because of the absence of nonlinearity in the key scheduling of the AES. Along these lines, a couple of optional key scheduling techniques have been put forward to oppose the key related assaults. In this segment, we survey these recommendations.

- Proposal 1: *meAES*

In [25], authors have proposed a different option to the existing key schedule of the AES. This proposition may be signified as *meAES*. The end point is to achieve a few wanted properties for AES key schedule, for example, collision resistant one-way function, insignificant common data and proficient implementation. Specifically, the primary feature is to accomplish irreversibility of the round key. This implies given any subset of the *meAES* round subkeys; it is challenging to infer the rest of the round subkeys. It should be noted that *meAES* was put forward before the most recent related-key assaults that figured out how to hypothetically break the full AES-192 and AES-256 out of 2009 [23].

In [26], the writers directed factual tests to demonstrate that their proposed scheduling of key used in *meAES* does not contain any bit spillage between round subkeys and enhanced confusion and diffusion of the bits by fulfilling the tests of frequency and Strict Avalanche Criterion (SAC).

- Proposal 2: *xAES*

The author in [27] put forward another scheme of key scheduling called *xAES*. The *xAES* is indistinguishable from the AES aside from the scheduling of key. Previously, the AES key schedule, with a specific end goal to get the values of a column in the W cluster, if the index of the column is a multiple of *Nk*, at that point, the past column is first rotated one byte up. In *xAES*, the rotation is carried out for all columns.

Moreover, *xAES* has other use of S-boxes for the key scheduling of xAES-192. A complete round of all *xAES* variations has protection against all fixed key assaults and key related differential assaults.

- Proposal 3: *imeAES*

Authors in [28] put forward an enhanced form of the meAES key calendar. The change is carried out to fix a small shortcoming found in the latter's key schedule.

Authors demonstrated that the *meAES* key scheduling has corresponding keys that create a similar encryption yield. The shortcoming is because of the initialization of *a* and *b*. In May et al.'s key timetable, a foe can compel *a* and *b* to have zero differential by picking a proper combination of keys.

The researchers adjusted initialization of *a* and *b* to such an extent that every byte of a and b relies upon one byte rather than two bytes from the key. They additionally presented a constant *keylen* depending on the length of the key to safeguard against the related cipher assaults [10]. In this case, *keylen* means the length of the key of the cipher which has been encoded as a byte. Same as the first form by May et al., this enhanced key scheduling likewise has the property of round key irreversibility since the structure is the same as *meAES*.

- Proposal 4: *ceAES*

Aside from enhancing the *meAES* key scheduling, authors in [29] likewise proposed another key scheduling for the AES. The proposition gives incomplete round key irreversibility. This is because, given a few mixes of at least three round subkeys, it might be conceivable to get the rest of the round subkeys. Assuming, in any case, two round subkeys are known, it is challenging to acquire the rest of the round subkeys.

Strengths:

- As far as effectiveness is concerned, the key scheduling of *xAES* is the best choice to supplant the actual AES key timetable. This is because of the negligible changes done to the actual scheduling of keys.
- The key scheduling schemes put forward are safe against the most recent key related assaults.

Limitations:

- The proposed *xAES* key schedule scheme results in performance degradation. The degradation of execution of these new key schedule recommendations is because of the presence of new transformations to the current AES key schedule.
- It might be legitimate to supplant the actual key scheduling of the AES with another one. In any case, this may affect numerous security applications that utilize the AES because of the progressions that should have been performed.
- No improvement in resistance against Brute force attack.

C. Open Issues

After reviewing the security enhancement techniques in Advanced Encryption Standard been proposed by scientists, it can be presumed that a few escape clauses still exist in the security arrangements put forth. The constraints have been observed to relate to a few fields from technical adoptability to computational complexity. At last, the dynamic overview directed in this paper closes with the induction of open issues as specified underneath:

- Avalanche effect needs to be improved.
- No measures have been taken to counter Traffic Analysis and monitoring.

The biclique [16] attack, based on the meeting-the-middle attack and is presently the most well-known attack on variations of all key sizes of the AES remains untouched.

V. SYSTEM DESIGN

Nowadays, information security is progressively becoming more critical since individuals have a considerable measure of individual information and system correspondences that should remain secure. This is the reason encryption of the data is required, and it is essential to ensure that ordinarily utilized encryption strategies are safe. Lamentably there is no real way to give completely consummate information security, yet it is conceivable to ensure that it is computationally infeasible to unscramble a scrambled message without the secret key. AES is considered to be the most secure cipher known so far, but there are some theoretical attacks against AES. Due to high computational complexity, these attacks are generally infeasible to execute. But, the power of computing of PCs is doubled after each one and a half years it would take just decades when AES turns out to be computationally uncertain. A few cryptographers remarked at a beginning period that the algebraic structure and outstanding effortlessness of AES might be a fountainhead of worry and years later, many theoretical attacks against AES were announced which threatens the confidentiality of information. In specific, the key related kind of assaults figure out how to theoretically break AES with 192-and 256-bit keys [26]. Another exceptionally new attack, the biclique [30], is guaranteed to compromise the AES in a single key assault model theoretically.

VI. PROPOSED SYSTEM

The proposed system is much more secure than the before mentioned approaches since it provides security and supports the performance with continued existence. The proposed algorithm to enforce AES modifies existing AES algorithm

with inclusion of dynamicity in AES S-Box to increase its immunity against the Brute force attacks & to increase its resistance against traffic analysis and monitoring. It also uses the dispersion of key over intermediate ciphertext to enhance the avalanche effect of AES Algorithm, i.e. Confusion and Diffusion. The dynamic character of AES S-Box is attained by using ten different S-Boxes with their corresponding inverse S-Boxes instead of a single S-Box. One of the ten S-Boxes is chosen on the basis of random number generated in the range 1-10 using pseudo-random number generation function to encrypt and decrypt the data/Message (e.g. If pseudorandom number generation function generates "3", the S-Box 3 will be used for encryption and decryption). The fundamental idea of random S-Boxes is to utilize S-Box in which each substitution step is free from any other. Now, each time the Message is converted into different cipher, thereby increasing resistance against traffic analysis and monitoring. Also, with the introduction of hashing technique in AES algorithm, variance in security can be achieved using MD4, SHA3 or SHA5.

A. *Functional Description of Modules*

The principal objective of the work being put forward is to design a model for Encryption and Decryption. The whole process can be effectively visualized with the help of modules which have been described as follows.

a) Encryption Module

The core purpose of the component is to encrypt the data, i.e. conversion of plain text into cipher text using the following steps of encryption:

- Generate a pseudo-random number {*r*} in the range 1-10.
- This random no. is then used for two purposes:
  i. Determining the S-Box to be used in Encryption and Decryption.
  ii. For generating Hash value.
- Use the selected S-box to perform encryption using the existing AES algorithm.
- Also, disperse the key on Intermediate Cipher text to generate the new Cipher text.
- The hash value and cipher text that is generated are sent to the receiver.

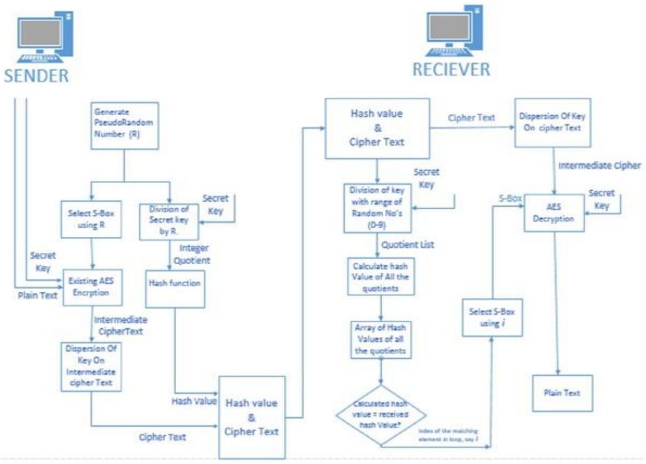

SENDER
RECIEVER
Generate PseudoRandom Number (R)
Select S-Box using R
Division of Secret key by R.
Secret Key
Integer Quotient
Secret Key
Plain Text
Existing AES Encrption
Hash function
Intermediate CipherText
Dispersion Of Key On Intermediate cipher Text
Hash Value
Hash value & Cipher Text
Cipher Text
Hash value & Cipher Text
Cipher Text
Dispersion Of Key On cipher Text
Intermediate Cipher
Secret Key
Division of key with range of Random No's (0-9)
Quotient List
Calculate hash Value of All the quotients
Array of Hash Values of all the quotients
Calculated hash value = received hash Value?
Index of the matching element in loop, say i
Select S-Box using i
S-Box
AES Decryption
Secret Key
Plain Text

Fig. 1. Proposed AES algorithm for encryption and decryption.

b) Decryption Module

The core purpose of this component is to decrypt the data, i.e. conversion of the cipher text into plain text using the following Decryption steps:

- The hash value that is received is used to determine the S-Box that was used for Encryption.
- Disperse the key on received Cipher text to generate the intermediate Cipher text.
- Use the selected S-Box to perform Decryption using the existing AES algorithm.

c) Selection of S-Box

A random generation function is used that generates a random number called Pseudorandom Number (*R*). Each of the S-Boxes is numbered from 1 to 10. The random number generated will determine the S-Box to be used for the particular transmission. For instance, if the random number generated is “5”, the S-Box numbered 5th is chosen to encrypt the message. Now, at the receiver end, the S-Box to be used should be known to decrypt the message correctly. Sending index number of the S-Box on the transmission line could be intercepted by the intruder; hence the hash value of the integer quotient of the Secret key divided by the random number is sent instead. Since the hash value is irreversible, data integrity & security is guaranteed.

Now, at the receiver end, for determining the S-Box to be used, the secret key is iteratively divided by the no’s (1-10) and the corresponding hash values are stored in an array. The hash value that is received at the receiver end is compared with the hash values in the array. The index of the match in the array will determine the s-Box to be used. For instance, if the hash value at the i_{th} index of an array is equal to the received hash value, then the *ith* s-box will be used for decryption.

d) Key Dispersion

In key dispersion step, the output that is obtained after Encryption by using proposed AES is further processed by dispersing the key on Intermediate Cipher text (i.e. the data which is obtained by encrypting the plain text using enhanced AES algorithm) to generate the new Cipher text. This is done to increase the Avalanche effect AES Algorithm. Avalanche effect is a critical standard for any encryption calculation. This property is observed while transforming one bit in plaintext and afterwards observing the impact of that change in the result of at least half of the bits in the cipher text. One reason for the avalanche effect is that by changing just a single piece, there is extensive change then it is harder to play out an investigation of the cipher when endeavouring to think of an assault. Therefore, the key dispersion technique is introduced to increase the avalanche effect and hence increase the immunity of the AES.

VII. DESIGN IMPLEMENTATION

The implementation details have been highlighted, starting with how to install and set everything up. The JDK comes with all the tools and APIs necessary to write Java apps. All that's left to do is to set it up properly. We need the Java Development Kit (JDK) and an Editor or preferably an Integrated Development Environment (IDE) for Java-like Eclipse to make development easier and more comfortable.

The following order of installation is recommended:

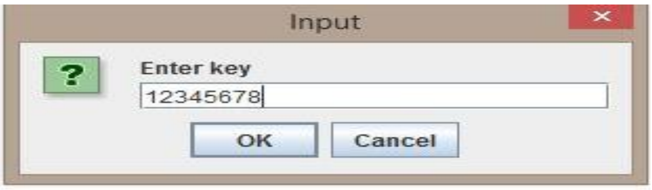
1. Java Development Kit
2. Eclipse

A. *Illustration of the Proposed System*

The screenshots that follow illustrate the entire process of Encryption and Decryption in a stepwise manner are as follows:

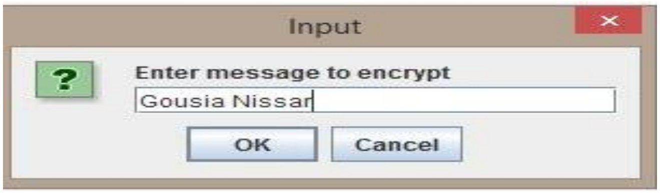
1) Encryption

The implementation details have been highlighted, starting with how to install and set everything up. The JDK comes with all the tools and APIs necessary to write Java apps. All that's left to do is to set it up properly. We need the Java Development Kit (JDK) and an Editor or preferably an Integrated Development Environment (IDE) for Java-like Eclipse to make development easier and more comfortable.



Enter key
12345678

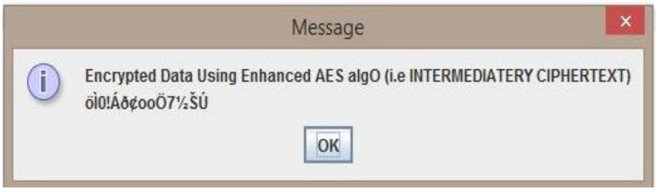
Fig. 2. Screenshot of the page asking the user to enter the private key.



Enter message to encrypt

Gousia Nissar

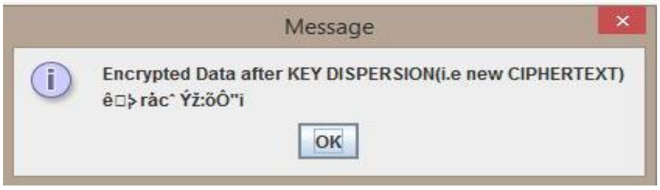
Fig. 3. Screenshot of the page asking the user to enter the data.



Encrypted Data Using Enhanced AES algO (i.e INTERMEDIATERY CIPHERTEXT)

öÌ0!Áð¢ooÖ7½ŠÚ

Fig. 4. Screenshot showing intermediate cipher text.



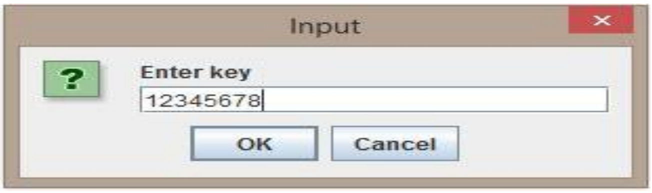
Encrypted Data after KEY DISPERSION(i.e new CIPHERTEXT)

ê□þràcˆ Ýž:őÔ"i

Fig. 5. Screenshot showing encrypted data an actual cipher text.

2) Decryption

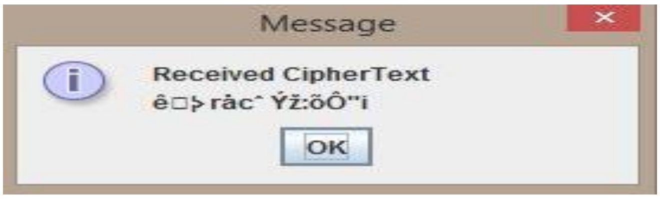
To decrypt the received cipher, the user has to enter the shared secret key. The received ciphertext is displayed via a dialog box. The encrypted message (real ciphertext) is processed by the key dispersion module, and the intermediate ciphertext is obtained and displayed via a dialog box. The intermediate cipher is then decrypted using the proposed enhanced AES algorithm, and the message (Plain Text) is retrieved and displayed via dialog box.



Enter key

12345678

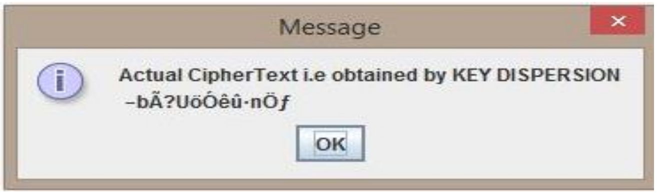
Fig. 6. Screenshot of the page asking the user to enter the private key.



Received CipherText

ê□þràcˆ Ýž:őÔ"i

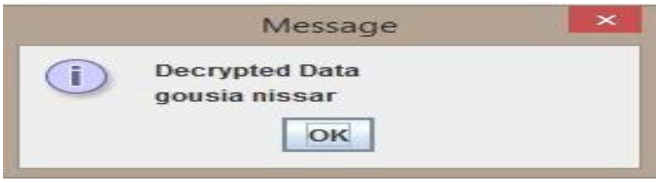
Fig. 7. Screenshot of received cipher text.



Actual CipherText i.e obtained by KEY DISPERSION

–bÃ?UöÓêû·nÖƒ

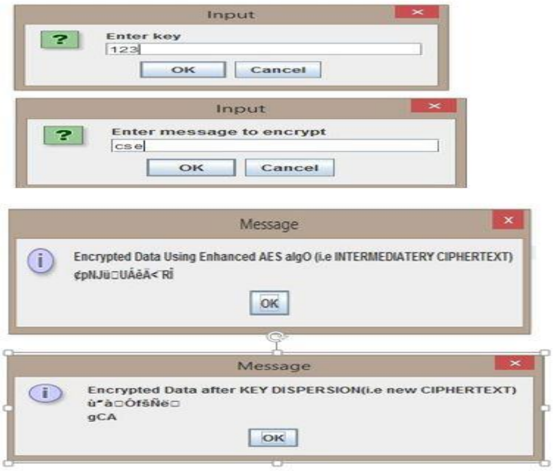
Fig. 8. Screenshot of actual cipher text.



Decrypted Data

gousia nissar

Fig. 9. Screenshot of decrypted data.



Enter key

123

Enter message to encrypt

cse

Encrypted Data Using Enhanced AES algO (i.e INTERMEDIATERY CIPHERTEXT)

¢pNJü□UÁëĀ<RÎ

Encrypted Data after KEY DISPERSION(i.e new CIPHERTEXT)

ù˝à□ÓfšÑë□
gCA

Fig. 10. Screenshot of credentials for data encryption.

VIII. ANALYSIS

We present the results of the AES algorithm put forward; the results are compared with the existing AES Algorithm. We discuss the improved resistance of AES Algorithm against Brute Force Attack, show how the probability of traffic analysis and monitoring is reduced, and finally, the improvement in Avalanche effect due to Key dispersion Technique is discussed.

A. *Resistance against Biclique Attack*

In the meet-in-the-middle (MITM) assault, the assailant needs matches of plaintext and relating ciphertext. The aggressor partitions the cipher into two sub ciphers. One of the sub-ciphers encodes the plaintext, and alternate unscrambles the comparing cyphertext. The thought is to make these sub ciphers to "meet in the middle" by looking for a right key-match. This technique is ineffectual against AES because it has a nonlinear key schedule. MITM assault can be improved by using a biclique structure. Complete bipartite graph or Biclique comprises of two sets of vertices in which each vertex of the principal bunch is associated with all vertices in the second bunch. The most efficient cryptanalysis against AES is the Biclique attack. For 128-bit key, it has a computational complexity of $2^{126.18}$. The proposed algorithm increases the complexity to $2^{126.18\times10}$, making it almost impossible to break the key using Biclique attack.

B. *Resistance against Brute Force Attack*

The brute force assaults are fundamental assaults, and they can be utilized against every cipher. With these assaults, the objective is to experience each conceivable key change until the point that it finds the key that can unravel the information into plaintext [30]. In most dire outcome, the brute force must attempt each key change before the right one is found.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)

For 128-bit key of AES, the worst-case time is 2^{128}. No Practical Attacks have been announced yet that has broken full AES because of its high computational complexity. But two developing advancements have demonstrated their capacity in the brute force assault of specific ciphers. One is present-day GPU innovation; the other is the field programmable gate array (FPGA) innovation. GPUs advantage from their wide accessibility and value execution advantage, FPGAs from their vitality proficiency per cryptographic operation. The two advances provide the benefits of parallel processing to brute force assaults. If there should be an occurrence of GPUs nearly hundreds, on account of FPGA somewhere in the range of thousand handling units, improving them to be much suited to breaking passwords than regular processors.

Now, with the inclusion of Dynamic characteristic in S-Box, along with the key guessing, the attacker also has to guess the S-Box used. For every combination of 128-bit key, it has to check the 10 S-Boxes to decrypt the cipher correctly. It increases the resistance against Brute Force attack by the factor of 100.

Worst-case Complexity for existing AES (128-bit key) = 2^{128}

Worst-Case Complexity for enhanced AES (128-bit key) $=2^{128}\times 10$

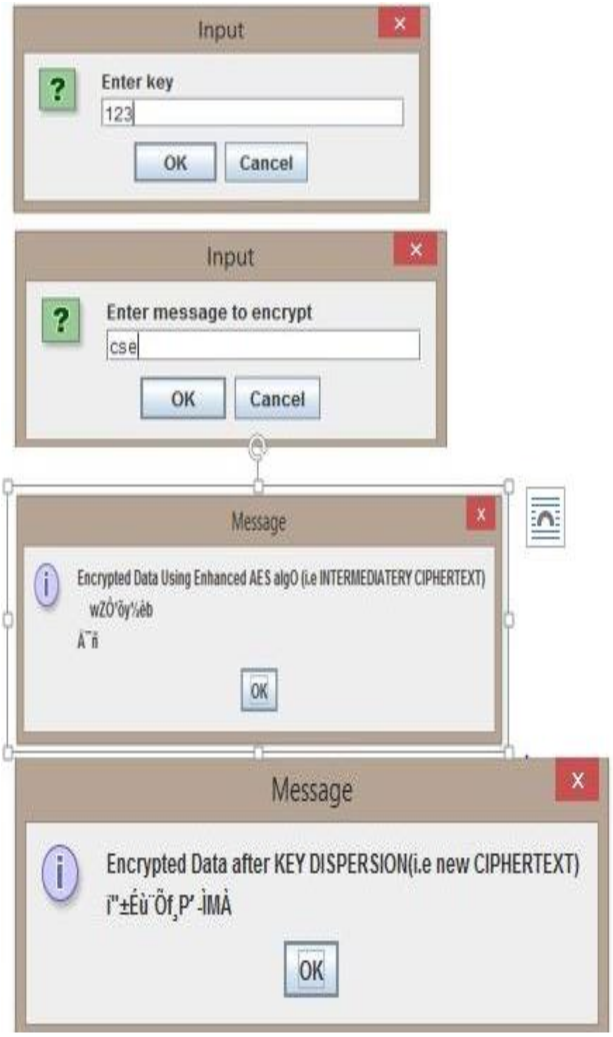
C. *Traffic Analysis & Monitoring*

System Traffic Monitoring is the way toward auditing, breaking down and overseeing system activity for any anomaly or processes that can influence the execution of the system, accessibility or the security. It is a system administration process that utilizes different devices and strategies to contemplate computer network-based packet traffic/information/communication. With the use of Single S-Box in AES, Traffic analysis was quite simple as a message was always encrypted to the same cipher.

But in the proposed AES algorithm, 10 S-Boxes are used. Each time the same Message is encrypted to the different cipher text with the use of different s-box and randomness in S-Boxes. This makes traffic analysis and monitoring nearly impossible. This is illustrated in the screenshots below; the same key and plain text are entered as input, and it generates the different ciphertext each time:

FIRST TIME: KEY=123; PLAIN TEXT=CSE

SECOND TIME: Same key and data, i.e. KEY=123; PLAIN TEXT=CSE



Enter key
123

Enter message to encrypt
cse

Encrypted Data Using Enhanced AES algO (i.e INTERMEDIATERY CIPHERTEXT)
wZÔ'õy½èb
Ã¯ñ

Encrypted Data after KEY DISPERSION(i.e new CIPHERTEXT)
ï"±Éù¨Õf¸P' -ÌMÀ

Fig. 11. Screenshot of credentials for data encryption.

D. *Improvement in Avalanche Effect Algorithm*

Avalanche effect is a critical standard for any encryption calculation. This property is observed while transforming one bit in plaintext and afterwards observing the impact of that change in the result of at least half of the bits in the cipher text. One reason for the avalanche effect is that by changing just a single piece, there is extensive change then it is harder to play out an investigation of the cipher when endeavoring to think of an assault. Therefore, the key dispersion technique is introduced to increase the avalanche effect and hence, the increase in immunity of the AES Algorithm against attacks.

E. *Algorithm*

Step 1. Create an array of cipher text.

Example

Cipher text, c = 1010000000−−−000010

1	0	1	0	0	0	0	0	1	0
0	1	2	3	4	5	6	7	126	127

Step 2. Create an array of key (128 bits).

Example
Key, k = 1010000000−−−000010

1	0	1	0	0	0	0	0

1	0

Step 3. Compute the new cipher text bits.

As for calculating the i^{th} bit of cipher text: Take the i^{th} bit of intermediate cipher text, i.e. $c[i]$.

From 128 bit key and using the random number (n), starting from ((1+n)%$k.length$) take i number of bits alternatively.

Then, from these bits perform even parity check. The result of the check will determine the new cipher text bits.

Example:

Random number; n = 3

Intermediate Cipher text;
C = 10110001

Key; k =10101110

New cipher text;

$C^0 = R1\ R2\ R3\ R4\ R5\ R6\ R7\ R8$

$R1 = c[1]k[(1+n)\%8]k[(3+n)\%8]k[(5+n)\%8]k[(7+n)\%8]$

$= c[1]k[(1+3)\%8]k[(3+3)\%8]k[(5+3)\%8]k[(7+3)\%8]$

$= c[1]k[4]k[6]k[8]k[2]$

$= \underbrace{10100}$

Even number of one's, so $R1$ = 0.

$R2 = c[2]k[(1+3)\%8]k[(2+3)\%8]k[(5+3)\%8]k[(6+3)\%8]$

$= c[2]k[4]k[5]k[8]k[1]$

$= \underbrace{00101}$

Even number of one's, so $R2$ = 0.

$R3 = c[3]k[(1+3)\%8]k[(2+3)\%8]k[(3+3)\%8]k[(7+3)\%8]k[(8+3)\%8]$

$= c[3]k[4]k[5]k[6]k[2]k[1]$

$= \underbrace{101101}$

Even number of one's, so $R3$ = 0.

$R4 = c[4]k[4]k[5]k[6]k[7]$

$= 10111$

$R4 = 0$

$R5 = c[5]k[4]k[5]k[6]k[7]k[8]$

$= 001110$

$R5 = 1$

$R6 = c[6]k[4]k[5]k[6]k[7]k[8]k[1]$

$= 0011101$

$R6 = 0$

$R7 = c[7]k[4]k[5]k[6]k[7]k[8]k[1]k[2]$

$= 00111010$

$R7 = 0$

$R8 = c[8]k[4]k[5]k[6]k[7]k[8]k[1]k[2]k[3]$

$= 001110101$

$R8 = 0$

Hence, New cipher text (i.e. send to receiver) is

$C^0 = R1\ R2\ R3\ R4\ R5\ R6\ R7\ R8$

$\therefore C^0 = 00001000$

As evident from the formations above each bit value within the cipher text is determined by the parity check of corresponding intermediate cipher text bit and four distinct key bits selected on the pattern of Hamming code. Any key bit, say $k[1] - -k[8]$ determines the output of one to eight cipher text bits as deliberated below:

$k[1]$→5 cipher text bits
$k[2]$→4 cipher text bits
$k[3]$→1 cipher text bit
$k[4]$→8 cipher text bits
$k[5]$→7 cipher text bits
$k[6]$→7 cipher text bits
$k[7]$→5 cipher text bits
$k[8]$→6 cipher text bits

On an average, this means that one key bit change would affect 43⁄8 = 5.37 bits in the cipher text. This means the avalanche effect is 5.37/8 = 67.18%. Thus, the key dispersion technique used increases the avalanche effect from 46%. (Avalanche effect of Original AES) to 67.18% (Avalanche effect of proposed AES).

IX. CONCLUSION

The algorithms related to encryption assumes a critical part in the security of communication. This examination work studied the execution of existing encryption procedures like AES, DES and RSA. We came to know that Encryption and Decryption of the AES algorithm is superior to the others in terms of performance, security and speed. In this study, we have described efficient extensions (such as Dynamic characteristic of S-Box, Key Dispersion Technique, and hashing technique) of AES to make it up to the level in terms of security with the evolution in technology. The Experiment results show that our algorithm, is slower and has more complexity than Original AES, but we can ignore this for two reasons: first that the increase in time and complexity in our algorithm can be neglected especially when we are using computers nowadays. The second to get the benefits that our algorithm makes the new attacks on symmetric key cipher extremely difficult. It is expected (if no genuine shortcomings of this approach are discovered) it might give a constructive alternative to the current implementations of AES.

The algorithm is rather striking and requires additional consideration. However, so far, there are no shrouded shortcomings embedded to cover the blemishes.

ACKNOWLEDGMENT

We would also like to show our gratitude to Mr M. Mueen Ul Islam Mattoo, Ms Rifat Gulshan and Mr Umar Hurrah for sharing their pearls of wisdom with us during the course of this research. We are also immensely grateful to anonymous reviewers for their comments on an earlier version of the manuscript, although any errors are our own and should not tarnish the reputations of these esteemed persons.

REFERENCES

1. M. Masihuddin, B.U.I. Khan, M. Mattoo and R. Olanrewaju, "A Survey on E-Payment Systems: Elements, Adoption, Architecture, Challenges and Security Concepts", Indian Journal of Science and Technology, vol. 10, no. 20, pp. 1-19, 2017.
2. Biryukov, D. Khovratovich, and I. Nikolic, "Distinguisher and related-key attack on the full aes-256", Advances in cryptology-crypto 2009, Springer, 2009, pp. 231–249.
3. B.U.I. Khan, R. Olanrewaju, F. Anwar, A. Najeeb and M. Yaacob, "A Survey on MANETs: Architecture, Evolution, Applications, Security Issues and Solutions", Indonesian Journal of Electrical Engineering and Computer Science, vol. 12, no. 2, pp. 832-842, 2018. Available: 10.11591/ijeecs.v12.i2. pp832-842
4. S. Hussain, B.U.I. Khan, F. Anwar and R. Olanrewaju, "Secure Annihilation of Out-of-Band Authorization for Online Transactions", Indian Journal of Science and Technology, vol. 11, no. 5, pp. 1-9, 2018.
5. T. Mehraj, B. Rasool, B.U.I. Khan, A. Baba and A. Lone, "Contemplation of Effective Security Measures in Access Management from Adoptability Perspective", International Journal of Advanced Computer Science and Applications, vol. 6, no. 8, pp. 188-200, 2015. Available: 10.14569/ijacsa.2015.060826.
6. B. Rasool, T. Mehraj, B. Khan, A. Mehraj and Z. Najar, "Securely Eradicating Cellular Dependency for E-Banking Applications", International Journal of Advanced Computer Science and Applications, vol. 9, no. 2, pp. 385-398, 2018. Available: 10.14569/ijacsa.2018.090253.
7. W. Stallings, Cryptography and network security: Principles and practices. Pearson Education India, 2006.
8. Announcing the Data Encryption Standard, Federal Information Processing Standards Publication 46-3, pp. 1-27, 1999.
9. H. Feistel, "Cryptography and computer privacy," Scientific american, vol. 228, no. 5, pp. 15-23, 1973.
10. Announcing the Advanced Encryption Standard (AES), Federal Information Processing Standards Publication 197, pp. 1-51, 2001.
11. B.U.I. Khan, R. Olanrewaju, A. Baba, S. Lone and N. Zulkurnain, "SSM: Secure-Split-Merge Data Distribution in Cloud Infrastructure", in 2015 IEEE Conference on Open Systems (ICOS), Melaka, Malaysia, 2015, pp. 40-45.
12. H. Gilbert and M. Minier., "A Collision Attack on 7 Rounds of Rijndael", in The Third Advanced Encryption Standard Candidate Conference, 2000, pp. 230–241.
13. N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting, "Improved Cryptanalysis of Rijndael", in Bruce Schneier, editor, Fast Software Encryption: 7th International Workshop, Springer-Verlag, 2001, pp. 213–230.
14. H. Demirci and A. A. Selc¸uk, "A Meet-in- the-Middle Attack on 8-Round AES", in Kaisa Nyberg, editor, Fast Software Encryption: 15th International Workshop, FSE 2008, vol. 5086 of Lecture Notes in Computer Science, Springer-Verlag, 2008, pp. 116–126.

A. Bogdanov, D. Khovratovich, and C. Rechberger, "Biclique Cryptanalysis of the Full AES", in Dong Hoon Lee and Xiaoyun Wang, editors, Advances in Cryptology- ASIACRYPT 2011 - 17th International Conference on the Theory and Application of Cryptology and Information Security, vol. 7073 of Lecture Notes in Computer Science, Springer-Verlag, 2011, pp. 344–371.

15. H. O. Alanaji, A. A. Jaidan, B. B. Jaidan, H. A. Jalab and Y. Al-Nabani, "New Comparative Study Between DES, 3DES, AES Within Nine Factors.," Journal of Computing, vol. 2, no. 3, pp. 152-157, 2010
16. G. Jakimoski and Y. Desmedt, "Related-Key Differential Cryptanalysis of 192-bit Key AES Variants", in M. Matsui and R. J. Zuccherato, editors, Selected Areas in Cryptography, 10th Annual International Workshop, SAC 2003, vol. 3006 of Lecture Notes in Computer Science, Springer-Verlag, 2004, pp. 208–221.
17. N. Ferguson, J. Kelsey, S. Lucks, B. Schneier, M. Stay, D. Wagner, and D. Whiting, "Improved Cryptanalysis of Rijndael", in B. Schneier, editor, Fast Software Encryption: 7th International Workshop, FSE 2000, vol. 1978 of Lecture Notes in Computer Science, Springer-Verlag, pp. 213–230, 2001.
18. E. Biham, O. Dunkelman, and N. Keller, "Related-Key Boomerang and Rectangle Attacks", in R. Cramer, editor, Advances in Cryptology - EUROCRYPT 2005, vol. 3494 of Lecture Notes in Computer Science, pp. 507–525. Springer-Verlag, 2005.
19. J. Kim, S. Hong, and B. Preneel, "Related-Key Rectangle Attacks on Reduced AES-192 and AES-256", in A. Biryukov, editor, Fast Software Encryption: 14th International Workshop, FSE 2007, vol. 4593 of Lecture Notes in Computer Science, pp. 225–241. Springer-Verlag, 2007.

A. Biryukov, O. Dunkelman, N. Keller, D. Khovratovich, and A. Shamir, "Key Recovery Attacks of Practical Complexity on AES Variant with Up To 10 Rounds", IACR eprint server, 2009/374 July 2009, http://eprint. iacr.org/2009/374.

20. F. Standaert, G. Rouvroy, J. Quisquater, and J. Legat, "Efficient implementation of Rijndael encryption in reconfigurable hardware: Improvements & design tradeoffs," in Proc. CHES 2003, Cologne, Germany, 2003.
21. 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.

A. Janadi and D. A. Tarah, "AES Immunity Enhancement against algebraic attacks by using dynamic S-Boxes", in the proceedings of 3rd International Conference on Information and Communication Technologies: From Theory to Applications, 2008.

22. L. May, M. Henricksen, W. Millan, G. Carter, and E. Dawson, "Strengthening the key schedule of AES", in the proceedings of 7th Australasian conference on Information Security and Privacy - Lecture Notes in Computer Science, Vol. 2384, pp. 226-240, 2002.
23. Biryukov and D. Khovratovich, "Related-key cryptanalysis of the full aes-192 and aes-256," Advances in Cryptology– ASIACRYPT 2009, pp. 1–18, 2009.
24. N. Stoinov, "One Approach of using Key-Dependent S-BOXes in AES", in the proceedings of 4th International conference, Multimedia communications, Services and Security - Communications in computer and information science, Vol. 149, pp. 317- 323, 2011.
25. J. Choy, A. Zhang, K. Khoo, M. Henricksen and A. Poschmann, "AES Variant Secure Against Related-key, Differntial and Boomerang attacks", in the proceedings of 5th International workshop on Information Security Theory and Practice. Security and Privacy of Mobile Devices in Wireless Communication - Lecture Notes in Computer Science, Vol. 6633, pp. 191-207, 2011.
26. L. Hathaway, "National policy on the use of the advanced encryption standard (aes) to protect national security systems and national security information," National Security Agency, vol. 23, 2003.

A. C. Paar and I. J. Pelzl, "Introduction to Public-Key Cryptography," in Understanding Cryptography, Springer Berlin Heidelberg, 2010, pp. 149-171.

AUTHORS PROFILE

Gousia Nissar is pursuing M. Tech degree in Computer Science Engineering at Sri Sai College of Engineering and Technology, Badhani Pathankot.She received B.Tech. in CSE from Islamic University of Science and Technology, Kashmir, in 2016. Her research areas of interest are Information Security, Security algorithms, Wireless Communication and Networks.

Dr Dinesh Garg is a Professor & HoD in the Department of Computer Science & Engineering at SSCET Badhani (Pathankot) Gurdaspur, Punjab India. He received B.Tech. in IT from Kurukshetra University, India in 2007, M.Tech. in IT from Mharishi Markandeshwar University, Mullana India in 2010 and PhD from Banasthali Vidyapaith, India in 2017.

Burhan Ul Islam Khan is a PhD Scholar and Teaching Assistant at the Department of Electrical & Computer Engineering, International Islamic University Malaysia. He received B.Tech. in CSE from IUST, Kashmir, and MS in CIE from IIUM, Kuala Lumpur during 2011 and 2014 respectively. Before commencing his Ph.D., he has been involved in varying roles as that of Software Engineer, Research Analyst and Assistant Professor. His current research interests include Formulation of Bio-Inspired optimization models in IOT, Designing One Time Password Schemes, Employing Mechanism Design, and Game theory to protect ad-hoc networks.

Published By:
Blue Eyes Intelligence Engineering
and Sciences Publication (BEIESP)