International Journal of Science and Research (IJSR)
ISSN: 2319-7064
SJIF (2020): 7.803

Review of Symmetric Cryptographic Ciphers

Anshumaan Chauhan[1], Ayushi Agarwal[2]

BITS Pilani, Dubai, United Arab Emirates

Abstract: ***Cryptography is a field of computer science which deals with the integrity and confidentiality of the data sent across the internet. It converts the data into a meaningless form, from which the attacker will not be able to infer any useful information. One of the most used types of cryptography for data hiding is symmetric key cryptography. In this paper, we have given a summarized form of all the reviews done on symmetric key cryptographic algorithms along with their advantages and disadvantages. We have also proposed a solution that helps us to overcome the disadvantage of the AES cryptographic algorithm.***

Keywords: Authentication, Confidentiality, Data Encryption and Decryption, Integrity, Security

1. Introduction

Cryptography is one of the most challenging as well as important field in computer science. Security of data and information is given utmost importance. As the time passes and the computational power of technology increases, old cryptographic techniques must be replaced with something new. This has forced many organizations to question their cryptographic algorithm.

Cryptography is a Greek word, which is combination of 2 words: "crypto" and "graphy", which means hidden writing, or we can interpret it as data written in a secured form. Crytography is a field which deals with creation of new ciphers which can guarantee 4 basic principles: Authentication, Confidentiality, Integrity and Non-Repudiation.

Authentication means that when data is to be sent from A to B, then A should confirm its identity to B only then B accepts the sent packet. It should not be like data is sent by C but B thinks that A was the sender. When there is any change in the sent data, due to some unreliable channel or by the attacker, then receiver should know that packet has been tampered and should not accept it, this is called *Integrity*. *Non-repudiation* is the case where the sender refuses to accept that he/she is the sender of the packet received by receiver. *Confidentiality* refers to a situation, when data sent from A to B, can only be read by A and B, and even if any attacker is able to get the data, then he should not be able to make sense out of it.

Problem of confidentiality can be solved by cryptography in 3 ways: Symmetric key cryptography (also called private key cryptography), Asymmetric key cryptography (public key cryptography) and Hash functions.

Symmetric key cryptography makes use of only 1 key for both encryption as well as decryption purpose. In figure 1 the working of symmetric key cryptography is shown.

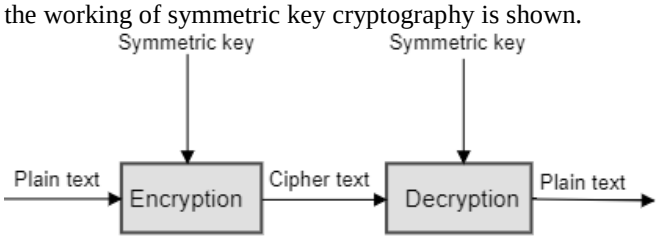

Figure 1: Private-key Cryptography

Algorithms like DES, AES, RC6, Blowfish, etc. comes under the category of symmetric key cryptography.

Asymmetric cryptography makes use of 2 keys, private key, and public key for secured data transfer. Data is encrypted using the public key of receiver at the sender's end and is decrypted using private key of receiver at receiver's end. Figure 2 shows the working of public key cryptography.

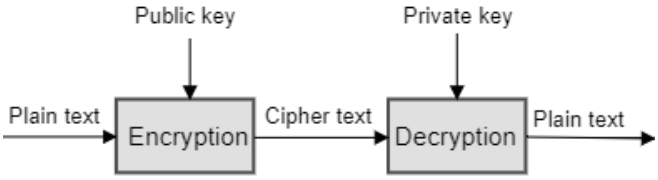

Figure 2: Public-key Cryptography

Due to use of 2 keys for encryption and decryption, it is slower than symmetric key cryptography. Algorithms such as RSA, SSL, etc. comes under this category.

Hash functions are also called one-way encryptors, they are used to encrypt the data without any key, it uses a non-reversible mathematical formula to encrypt the data. Some examples of hash functions are SHA-1 and MD5.

Figure 3 represents a flow chart of different types of encryption techniques.

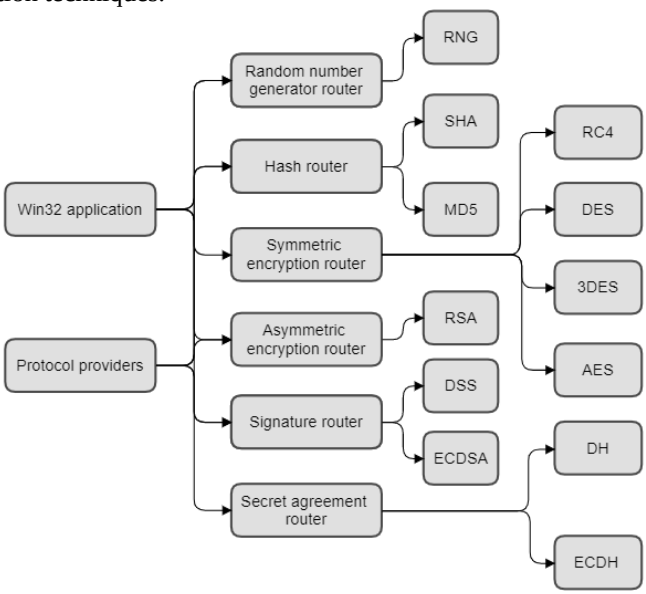

Figure 3: Types of Encryption Techniques

In section II of this paper the literature review or related work has been discussed, what were the problems that came up and a tentative solution is discussed in section III. Finally, conclusion is stated in Section IV.

2. Literature Survey

While doing symmetric key cryptography there are many challenges that are not concerned with the possibility of attacks an intruder can do by any cryptanalytic method. Some of these challenges involve which random key generation method to use for generating key sequence, key management, certificate revocation issue, and many more. There are many types of key generation techniques, such as True random number generator (TRNG), Pseudo random number generator (PRNG), cryptographic random number generator. Out of this cryptographic random number generator is mostly used. In terms of complexity TRNG is the best, but it is impossible to use in real life applications. That is because a random number cannot be transferred to the receiver over internet in a secured manner, unless there is a secured channel present. And if there is a secured channel ,then we can directly pass the data through it instead of sending it after encryption over an unsecured channel. In case PRNG, the key sequence can easily be known to the attacker, therefore it is also not used.

Certificate revocation is a situation which occurs when we use asymmetric key cryptography and the private key of the sender gets stolen/ attacker gets access to it in some unethical manner. Now the digital certificate is passed on to the receiver, and they verify identity of each other. But if some data is sent by receiver encrypted using public key of the sender, then attacker will also be able to decrypt it using sender's private key. In this case, sender adds his certificate issued by certificate authority in the revocation list, and requests to have issued a new certificate.

Satyabrata et al. [1] came up with a new approach in symmetric key cryptography foe wireless communication which used the concept and rules of cellular automata. The sued cryptographic technique uses a single block of 1-D PCA applied in WSN. The proposed method was prone to brute force as well as cryptanalytic attacks. The implementation complexity and memory consumption were also less.

Jissy Ann et al. [2] stated different types of cryptographic techniques used in cloud to prevent data, such as identity-based encryption, Attribute based encryption, and many more. It stated that in symmetric key algorithms there was an inverse proportion relation between size of the input file and size of the key (except for RSA). It concluded AES as the best algorithm, but the blowfish was the best in terms of time. It also mentioned that symmetric or asymmetric key algorithms were not sufficient alone for cloud security.

Nivedita et al. [3] did a comparative study on both asymmetric and symmetric key cryptographic algorithms. For analysis, they considered factors such as key length, speed, tunability, power consumption, security, cost, and implementation. After the comparative study they concluded that symmetric key cryptographic algorithms are superior to asymmetric ones in terms of speed and power consumption, whereas asymmetric ones are better in terms if tunability.

Sourabh Chandra et al. [5] stated advantages and disadvantages of symmetric cryptographic algorithms after comparing them based on 8 methods, which are as follows: Introduction of chips/cards and their authentication processes, Oblivious Transfer and Oblivious Attribute Certificates, Public key certification and revocation, Watering Scheme and Image Authentication, Production of data in cloud, Symmetric key encryption algorithm based on linear geometry, Symmetric key encryption algorithm based on elliptic curve and chaotic system and secure protocol using Quantum wave function. After considering all drawbacks and advantages of each system, they found digital watering scheme and public key certification and revocation were the most efficient mechanisms.

Abhishek Bhardwaj et al. [7] stated that there is a lot of research going on in the field where biometrics is combined with cryptography for better data protection. It also included that many methods have been proposed lately, where data hiding was done using concepts such as histogram, center folding strategy and significant qubit. The challenge faced by AES, DES and TDES was that processing power of computers are growing in an exponential manner and this is the reason why these algorithms are starting to fail eventually. One solution provided in [7] was to use steganography along with cryptography.

Shivani et al. [8] in the proposed algorithm used to change the block size while encryption instead of keeping it of a fixed size. They also used LSB image steganography algorithm for data hiding purpose. Experimentation results showed that difference between the MSE of proposed algorithm and that of the existing techniques was approximately 0.05 for all the iterations.

Amril et al. [9] proposed an improvised proxy re-encryption method for symmetric key cryptography. It proposed a method in which the ciphertext is passed to semi trusted party, and it encrypts it to another level, without getting access to the plaintext. So first the sender converts the plaintext into an intermediate form using AONT (all or nothing transform) and now this is sent to the semi trusted party for encryption. In experiments it was proved that the proposed method was also secure against chosen plaintext attack.

Muneer Bani Yassein et al. [10] in his analysis stated that when the block size is bigger in block ciphers, then the time taken to encrypt the data is lesser than what it takes to encrypt the same using smaller block size. It also stated the drawback of using symmetric key cipher wherein we must share keys with other parties during the process. AES was concluded to be the best algorithm as it takes minimum time for execution as well as least memory resources are consumed. Unlike AES, RSA was the one which takes maximum time for encryption and consumed largest memory size.

Bavanath et al. [11] did a review of multimedia cryptographic algorithms, and compared algorithms DES, TDES, AES,

IDEA, RAS based on following factors- key length, rounds, block size, cipher type, speed, and security. AES was the algorithm with best security amongst all and was the fastest. She also added that bio cryptography is the best method to hide multimedia data from unauthorized access.

Abhishek et al. [12] proposed a bit level symmetric key cryptography, along with genetic algorithm and embedding logic. In the proposed method, decrypted text can only be incurred when we have the initial key. After experimentation, method was said to be free from attacks like brute force, cryptanalytic attacks and known plaintext attack. Data integrity is a must while using this algorithm, as when even a single bit of the ciphertext changes, receiver will never be able to get back the plaintext, therefore its better for him to check integrity of the ciphertext first and then go for decryption.

Chitra et al. [14] implemented a hybrid cryptography approach in which they even encrypted the symmetric key that is used for encryption and this leads to better security. Later they make use of image steganography and uses LSB algorithm, which changes the first bit of image pixel to the encrypted bit of the data. And later this image is sent to the receiver.

Abdalbasit et al. [16] pointed out some of the advantages of using asymmetric key cryptographic algorithms over symmetric key cryptographic algorithms ,which are as follows:

- Key distribution can be done on public channels and no third party gets access of the key ever (unlike symmetric key).
- It needs less storage as compared to symmetric key algorithms which need to create a new key every time they want to establish a secure channel for communication.
- Asymmetric key algorithms are more suitable in the case of open environment communication, particularly in the situation when both the partis have never interacted with each other.

Abhishek Anand et al. [19] after a brief study of all existing symmetric key algorithms, proposed an algorithm which has the characteristics of simplicity, efficiency, security, data integrity and flexibility. It was easier to implement but difficult to decode.

Faiqa et al. [21] compared the symmetric key and asymmetric key algorithms on the following performance metrics: key size, generation time for key, encryption and decryption time taken by algorithms for different file sizes. In it, AES took the longest time to generate a key of length 128 bits, but the time for encryption and decryption of data for different file sizes was surprisingly very less as compared to others. In the end it was concluded that asymmetric schemes are computationally more expensive as compared to symmetric schemes.

3. Problem and Possible Solution

Shweta et al. [13] proposed an algorithm which solves the problem of key generation and sharing the session key using symmetric key cryptography. In the security analysis they did not find any potential attack, to which the proposed algorithm is prone to. They tested the algorithm using OFCM in the backend which detects the possibility of any attack. It mentioned the reason why they do not use the same key for different transaction, and that is because this method is prone to replay or playback attack.

Mohammad Ubaidullah Bokhari et al. [15] reviewed all the classical as well as modern symmetric key cryptographic algorithms and stated the uniqueness as well the attacks to which those are vulnerable to. AES which was shown to be the best amongst all in all the previous experiments, is prone to Side channel attack as well as known plaintext attack. DES which was one of the most used encryption algorithms till 1990s was prone to brute force attack and linear and differential analysis.

Shivlal et al. [17] stated limitations of all the symmetric key algorithms. For Advanced Encryption Algorithm (AES) also known as Rijndael, they stated that with observed mathematical and statistical property it is vulnerable to attack. For blowfish, which is another very powerful encryption algorithm, they mentioned that with the use of weak keys in 4 rounds, it is exposed to differential attacks with large number of weak keys.

In [5] it was mentioned that some of the best mechanisms used for symmetric key cryptography is Symmetric key encryption algorithm based on elliptic curve and chaotic system. Shafali et al. [18] proposed a method which uses fractal functions and makes use out of the relation between Madelbrot and Julia functions for developing a non-transitional cryptosystem. Experiments led to the conclusion that using fractal function enables us to use the chaotic nature and the size of the key being very large also makes it impenetrable to brute force attack. One advantage of this algorithm is that the initial values of the parameter are highly important. Algorithm being sensitive to initial values, if they are changed even a little bit, the value of decrypted message changes completely.

Muhammad Aamir Panhwar et al. [20] after studying different symmetric and asymmetric key encryption algorithms, compared the algorithms on key used, throughput, encryption ratio, tunability, power consumption, key length, speed, and to which attacks are these algorithms prone to. AES was prone to chosen plaintext attack and known plaintext attack, whereas blowfish was prone to dictionary attack. One more issue with AES algorithm is that its not tunable, we cannot change much in the algorithm according to our needs.

Ilayaraja et al. [4] gave an algorithm by modifying Caesar cipher (also called as additive cipher). Advantages of the proposed method were as follows:

- Simple in nature.
- Easy to encrypt and decrypt- less resource consumption.
- Consider a tab or space as part of plaintext and encrypts it too.
- Case sensitive.

This proposed algorithm overcame the drawbacks of Caesar cipher. We are going to use similar approach in our proposed algorithm but adding some extra components to overcome the drawbacks.

Aswin et al. [6] proposed an algorithm to speed up the process of encryption as well as making the algorithm to be more secure. In this approach the hackers cannot even crack which type of message file was it. So breaking cipher text becomes impossible in effect. A comparison done on basis of time between 3DES, IDEA, CAST-128 and proposed algorithm shows that for the same size of file, proposed algorithm was much faster than the others. It also concluded that as the size of file increases throughput also keeps on increasing. When analyzed how many years will it require to break a 20-character ciphertext then 125,000,000,000,000,000,000 years was the value they got. We will try to incorporate this type of encryption with AES to improve its security and make it invulnerable to all types of attack.

As we have seen in Section III, that the problems related to AES are that it is prone to known plaintext attack, chosen plaintext attack and side channel attack. Following we have proposed an algorithm to overcome the issue of known plain text attack. We cannot overcome the side channel attack. Basically, a side channel attack is an attack in which the attacker tries to decrypt the text as they know the internal working, rather than using cryptanalytic attacks. Some of techniques that are in the research papers on how to get prevented from side channel attacks are :

- Eliminating the release of private information or making sure this information is unrelated to your private data.
- Power line conditioning and filtering to deter power-monitoring attacks as well as emitting a channel with noise.
- Blinding technique that serves to alter the algorithm's input into some unpredictable state rendering some or all the leakage of useful information.

But these are not practically possible, as if we do not provide the algorithm used by us to the attacker, then we cannot be sure of how good and secure our algorithm is. So, we cannot change anything in our implementation for preventing from side channel attack. It can be minimized by keeping every information such as cache table, timing information, etc. very confidential.

Due to the restriction that AES is not tunable, we are going to add extra components to it, instead of changing its original functionality and working. As of now it is not proved that AES (or any block cipher) is resistant to known plain text, but they believed so because nobody has broken it until now, but as the processing power of the computers are increasing exponentially, one day sooner or later AES will be prone to known plain text attack (so prevention is better than cure, therefore we have proposed a methodology to avoid the problem of known plaintext attack). For preventing AES from known plain text attack we are going to:

- Make use of all 3 different kind of AES instead of a singular one while sending the document. And for specifying which type of AES algorithm was used for which block and for uniquely identifying the block, we will add type of AES, and sequence number at the last in a simple encrypted format, which can also be verified by the user using the next point. This will make our algorithm a bit slower, but it is not a problem as currently amongst all, AES is the fastest and most secured encryption algorithm. By this for decrypting the whole plaintext, even in the known plaintext attack, he would have to guess a total of billions of keys, which with current computation power of computers will also require years of time, and maybe by then this algorithm will be replaced by some better algorithm.
- And alongside this we are also going to use data integrity features such as digest using hash functions, which helps to prevent us from active attacks.

4. Proposed Algorithm

The two parties communicating will be using Advanced Encryption Standard (AES) for encrypting and decrypting the text.

AES algorithm has 3 different number of rounds executed based on the length of key used as shown in the table below.

Table 1: Types of AES architecture

Length of the key (bits)	Number of rounds
128	10
192	12
256	14

There are basically 4 operations that are performed in AES round that are Byte substitution, Shift rows, mix columns and key addition. For all the rounds all these 4 operations are performed except the last block which does not perform mix columns operation.

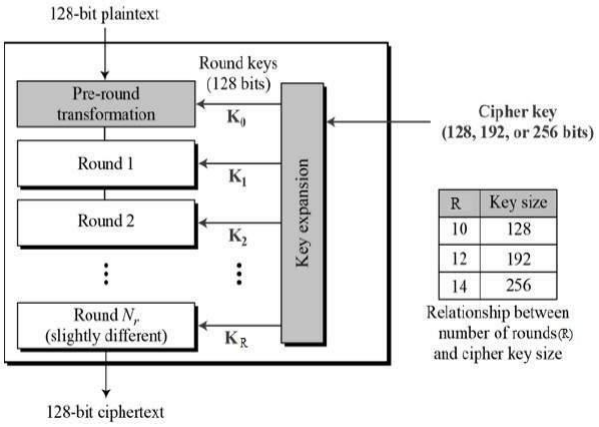

Figure 4: AES architecture

We are using ECB mode during encryption. Electronic code book (ECB) is a deterministic encryption mode. This is used for all block ciphers, and as we are also using a block cipher therefore we are choosing this as the mode of operation. Another advantage of using ECB mode is that we can easily incorporate the padding process in it when the length of plaintext does not match with the length of AES algorithm (128 bits). There is another mode of operation for block ciphers, i.e., Cipher block chain (CBC), but its drawback is that if there is error in anywhere in the block that will continue for all forthcoming blocks, and we don't want to have such a situation. ECB mode also has some drawbacks, based on the assumption that the key will not be changed for a long time, but the constraint of using the proposed method-

ology is that you must change the keys after every possible interval of time.

Encryption :

- Select the plaintext P to be encrypted.
- Check if the plaintext is of length multiple of 3, if not add filling character (can be a character that is mutually decided by both the parties earlier). If it is a multiple of 3 then divide it into 3 halves P_1,P_2,P_3.
- Create a symmetric key for each of the AES algorithm, i.e., 1 for AES when the key size is 128 bits, 1 when key size is 192 bits and 1 for key size of 256 bits. We have 3 keys k_1,k_2,k_3.
- Based on your choice (not fixing a strict pattern) encrypt one part of the plaintext with one encryption key. (Example- k_2 for P_2, k_1 for P_3 and k_3 for P_1) We get 3 ciphertexts as follows: C_1 , C_2 and C_3.
- Now add the bit size of the key used after the encrypted text of each block. (Will get a result of form like $C_1$256 , $C_2$192 , $C_3$128).
- Combine all blocks into one block. ($C_1$256$C_2$192$C_3$128)
- Create a new symmetric key k of size 128 bits.
- Encrypt this block with the newly created key and get the encrypted text C.
- Calculate the digest D of encrypted plaintext C using MD5 hashing algorithm.
- Send both digest D along with the encrypted plaintext to the receiver.

Decryption

- Get the ciphertext C and digest D.
- Calculate the digest D' of the Ciphertext C using MD5 hashing algorithm.
- Compare both the digests D and D'.
- If they are equal, that means no tampering was done and process with the following steps, but if they are different then discard the message received. This fulfills the purpose of ensuring integrity as well as preventing active attacks.
- Now using the symmetric key k decrypt the ciphertext C.
- Segregate the ciphertexts based on the position of values "128","192" and "256".
- Decrypt the ciphertexts wit the corresponding keys, i.e., one which was before 128 with k_1, ciphertext before 192 with k_2 and the one before 256 with k_3.
- Now we will have the three parts of the original plaintext P as P_1,P_2 and P_3.
- Combine them together to get P.

5. Simulation Results

The simulation of the proposed methodology was performed on BlueJ on Dell G3 15 laptop with the system specifications of I & intel 8th gen processor and 16 GB RAM. The plaintext P chosen was : "Crp Lab is interesting".

Encryption :

The length of the plaintext is 22. 22 is not a multiple of 3, closest multiple of 3 is 24. Common negotiated filling character is assumed to be '!'. Therefore, we have to add 24-22=2 filling characters at the end before starting the actual encryption process.

After adding of the filling characters, the new plaintext was : "Crp Lab is amazing!!".

Now this new P is divided into 3 halves, and the resulting P_1, P_2 and P_3 are shown in Figure 5.

```
Part 1 : Crp Lab
Part 9 : is inter
Part 17 : esting!!
```

Figure 5: Separated Plaintext

Symmetric keys used for encryption of these texts were as follows:

K_1 was used to encrypt P_1.
K_2 was used to encrypt P_2.
K_3 was used to encrypt P_3.

After the encryption we combined the data along with the key size of the key used to encrypt them on stage 1. The resulting plaintext that we get is shown in figure 6.

jJa!ÐÐpÐÐÐÐÐV1286Ð)Ðo&ÐÐÐ[Ð.ÐK192Ð)oEÐÐÐÐÐÐRe256Ð┐

Figure 6: New Plaintext after aggregating each individual encrypted part.

Symmetric key k is used to encrypt this combined new plaintext. The final ciphertext is:

Ð1Ð sJÐ ÐVÐÐbÐCqÐÐÐ]TÐ;ÐHÐSoTÐc ÐÐOε:ÐÐÐÐÐ ÐÐ gÐ6L ÐdᴬÐÐ Ð1Ð'ÐÐZÐÐSÐÐH z<Ðd)ÐÐsæDÐ ÐVÐWÐ\ÐÐÐÐ ÐÐ9ÐÐÐÐx'Ð)0

Figure 7: Final Ciphertext

Now we find the digest of this ciphertext using MD5 algorithm, we can also use SHA1, it depends on the negotiated terms between sender and receiver what hashing method we are using.

Digest of this plaintext was :

ÐÐ;ÐFGPÐúWÐИZ

Figure 8: Digest of the ciphertext

Now we assume that the text is transferred as it is from sender to receiver without any active attack done on it. Therefore, receiver receives digest and the encrypted text exactly what the sender sent i.e. it has not been tampered.

Decryption :

First receiver finds the digest using the hashing algorithm MD5 from the encrypted text. Now if this computed digest matches with the digest sent by the receiver then in that case, he will proceed with further decryption of the text, otherwise the message will be dropped there itself, as it signifies that tampering was done either in the text or in the digest sent. Here, both the digests will match since we assumed no tampering is being done.

Now the received ciphertext C is decrypted using symmetric key k. The resulting P' was :

jJa!ÐÐpÐÐÐÐÐV1286Ð)Ðo&ÐÐÐ[Ð.ÐK192Ð)oEÐÐÐÐÐÐRe256Ð┐

Figure 9: After first decryption of ciphertext using symmetric key k.

Based on the positions of the values, 128,192 and 256, receiver segregates the 3 different encrypted parts of the original plaintext P.
Receiver gets the following as C_1, C_2 and C_3.

When compare these and figure 6, we can see the match between the sent and received text.

Now according to the position of segregation and the value succeeding the text, appropriate key is used to decrypt the text.

In our case, C_1 is decrypted using k_1, C_2 using k_2 and C_3 using k_3.

Finally, receiver will get P_1,P_2 and P_3 . After appending them one after the other, original plaintext is available to the receiver.

```
Crp Lab
is inter
esting!!
Decrypted=Crp Lab is interesting!!
```

Figure 10: Individual separated plaintext and their corresponding aggregated text.

Performance that the proposed methodology gave is presented in Table 2.

Table 2: Performance measure of proposed algorithm

Word count	Encryption time (ns)	Decryption time (ns)
4	2642200	4881500
10	2355400	7044000
20	37771700	18428800
36	60398800	19263000

6. Security Analysis

1) Key space:
Advanced encryption algorithm was designed after Data Encryption Standard (DES) was discarded due to its drawback of having short length of the key. Key space of DES was 2^56 and was prone to brute force attack. Whereas key space of AES is 2^128, which not only prevents from Brute force attack (although there is a possibility of doing successful decryption using brute force attack with coming modern computers having high computational power), but also is resistant to differential attacks (DES is also resistant to differential attacks). But in our proposed methodology, the key space is 2^128+2^128+2^192+2^256, which makes it totally impossible for any computer to crack using brute force attack.

2) Key sensitivity:
Key is highly sensitive, even if one bit in the key changes, whole of the ciphertext will be different for the same plaintext. This is possible because of the diffusion and confusion concepts being applied using Permutation box and Substitution box within each round of the AES.

3) Known plaintext attack :
Possibility of known plaintext is near to impossible. Even if the attacker knew some of the plaintext, a proper decryption cannot be done because our first constraint was that key has to be changed at regular intervals. If we assume that we have not changed the key for a long period of time, equations cannot be solved because there is another layer of encryption happening, to which he does not know the plaintext for. So, the possibility of known plaintext is also zero.

4) Time complexity:
Figure 11 and 12 shows the graph specifying the time needed for encryption and decryption.

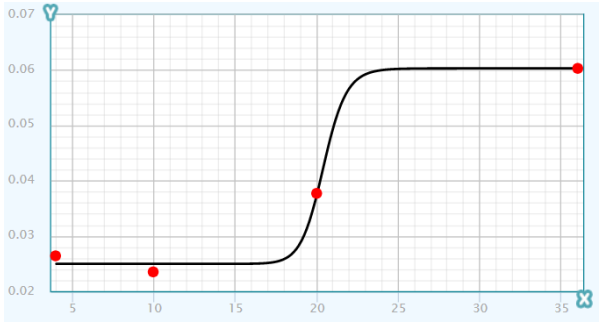

Figure 11: Time taken (s) v/s Number of words during encryption

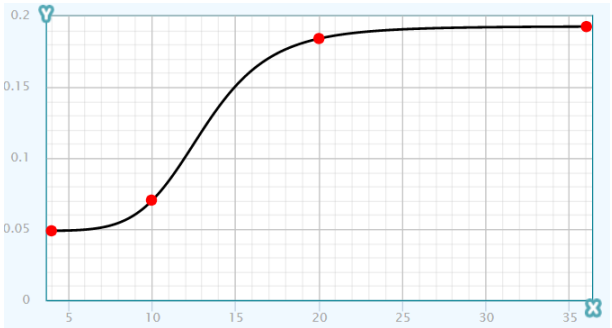

Figure 12: Time taken (s) v/s Number of words during decryption

Equations for the time complexity are as follows :
Encryption:

$$y = 0.0603 + (0.02495 - 0.0603)/(1 + (x/20.39595)^{29.19893}) \quad (1)$$

Decryption:

$$y = 0.1928032 + (0.0473243 - 0.1928032)/(1 + (x/13.06583)^{6.474961}) \quad (2)$$

Here y is Time (s) and x is number of words encrypted.

So we can see that if there is no tampering in the text and digest received by the receiver, the time complexity for both encryption and decryption is reasonable. If the data is tampered then, the time for verifying that it is tampered is even less and will be discarded much earlier.

7. Conclusion

Cryptography deals with changing data from one form to another meaningless from. There are many types of cryptographies existing, but the most used ones are asymmetric and symmetric key cryptographies. In this paper, we have summarized all the review papers done on the symmetric key cryptographic algorithms and found out the weakness of the most used algorithm in current days, that is Advanced Encryption algorithm (AES). AES is a block cipher that comes

under the category of symmetric key algorithms, and is prone to known plaintext attack, chosen plaintext attack and side channel attack. In the proposed methodology we have solved the problem of both known plaintext attack as well as future possible Brute force attacks by increasing the complexity of encryption and decryption, and also enhanced the algorithm in order to prevent from active attacks which helps us to incorporate integrity.

8. Future Scope

Confidentiality and integrity have been provided in the proposed algorithm. Now the future scope will be to solve the task of exchanging 4 keys to the receiver, using digital signature. Digital signature will make our algorithm a hybrid algorithm, as we will use symmetric key cryptography for encryption and decryption purpose, and public key cryptography for exchanging keys in a secured manner via Digital signature. We will encrypt the 4 keys using public key of the receiver and then we will send it to receiver along with the ciphertext. Receiver will decrypt those keys using its private key and then use them to decrypt the ciphertext.

References

[1] Parashar, Deepika, et al. "Symmetric key encryption technique: A cellular automata based approach." *Cyber Security*. Springer, Singapore, 2018. 59-67.

[2] George, Jissy Ann, and M. Hemalatha. "Cryptographic Techniques, Threats and Privacy Challenges in Cloud Computing." *Governance* 500: 2.

[3] Bisht, Nivedita, and Sapna Singh. "A comparative study of some symmetric and asymmetric key cryptography algorithms." *International Journal of Innovative Research in Science, Engineering and Technology* 4.3 (2015): 1028-1031.

[4] Ilayaraja, M., K. Shankar, and G. Devika. "A modified symmetric key cryptography method for secure data transmission." *International Journal of Pure and Applied Mathematics* 116.10 (2017): 301-308.

[5] Chandra, Sourabh, et al. "A study and analysis on symmetric cryptography." *2014 International Conference on Science Engineering and Management Research (ICSEMR)*. IEEE, 2014.

[6] Achuthshankar, Aswin, and Aswathy Achuthshankar. "A novel symmetric cryptography algorithm for fast and secure encryption." *2015 IEEE 9th International Conference on Intelligent Systems and Control (ISCO)*. IEEE, 2015.

[7] Bhardwaj, Abhishek, and Subhranil Som. "Study of different cryptographic technique and challenges in future." *2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*. IEEE, 2016.

[8] Chauhan, Shivani, Janmejai Kumar, and Amit Doegar. "Multiple layer text security using variable block size cryptography and image steganography." *2017 3rd International Conference on Computational Intelligence & Communication Technology (CICT)*. IEEE, 2017.

[9] Sakurai, Kouichi, Takashi Nishide, and Amril Syalim. "Improved proxy re-encryption scheme for symmetric key cryptography." *2017 International Workshop on Big Data and Information Security (IWBIS)*. IEEE, 2017.

[10] Yassein, Muneer Bani, et al. "Comprehensive study of symmetric key and asymmetric key encryption algorithms." *2017 international conference on engineering and technology (ICET)*. IEEE, 2017.

[11] Dhanalaxmi, Banavath, and Srinivasulu Tadisetty. "Multimedia cryptography—A review." *2017 IEEE International Conference on Power, Control, Signals and Instrumentation Engineering (ICPCSI)*. IEEE, 2017.

[12] Sen, Abhishek, Attri Ghosh, and Asoke Nath. "Bit level symmetric key cryptography using genetic algorithm." *2017 7th International Conference on Communication Systems and Network Technologies (CSNT)*. IEEE, 2017.

[13] Arora, Shweta, and Muzzammil Hussain. "Secure session key sharing using symmetric key cryptography." *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE, 2018.

[14] Biswas, Chitra, Udayan Das Gupta, and Md Mokammel Haque. "An efficient algorithm for confidentiality, integrity and authentication using hybrid cryptography and steganography." *2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)*. IEEE, 2019.

[15] Bokhari, Mohammad Ubaidullah, and Qahtan Makki Shallal. "A review on symmetric key encryption techniques in cryptography." *International Journal of Computer Applications* 147.10 (2016).

[16] Qadir, Abdalbasit Mohammed, and Nurhayat Varol. "A review paper on cryptography." *2019 7th International Symposium on Digital Forensics and Security (ISDFS)*. IEEE, 2019.

[17] Mewada, Shivlal, Pradeep Sharma, and S. S. Gautam. "Classification of efficient symmetric key cryptography algorithms." *International Journal of Computer Science and Information Security* 14.2 (2016): 105.

[18] Agarwal, Shafali. "Symmetric key encryption using iterated fractal functions." *International Journal of Computer Network & Information Security* 9.4 (2017).

[19] Anand, Abhishek, et al. "Proposed symmetric key cryptography algorithm for data security." *2016 International Conference on Innovation and Challenges in Cyber Security (ICICCS-INBUSH)*. IEEE, 2016.

[20] Panhwar, Muhammad Aamir, et al. "SACA: A Study of Symmetric and Asymmetric Cryptographic Algorithms." *International journal of computer science and network security* 19.1 (2019): 48-55.

[21] Maqsood, Faiqa, et al. "Cryptography: a comparative analysis for modern techniques." *International Journal of Advanced Computer Science and Applications* 8.6 (2017): 442-448.

Authors Short Profile

Anshumaan Chauhan is currently pursuing bachelors degree program in computer science engineering at BITS Pilani, Dubai, UAE, PH-+91-8979698899. E-mail: f20180274@dubai.bits-pilani.ac.in

Ayushi Agarwal is currently pursuing bachelors degree program in computer science engineering at BITS Pilani, Dubai, UAE, PH-+971-545489589. E-mail: f20180165@dubai.bits-pilani.ac.in