International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 04 | Apr 2021 www.irjet.net p-ISSN: 2395-0072

An Understanding and Perspectives of End-To-End Encryption

Ohwo Onome Blaise[1], Oludele Awodele[2], Odunayo Yewande[3]

[1-3]*Dept. of Computer Science, Babcock University, Illishan-Remo, Nigeria*

Abstract - *Digital communications surveillance is a major security concern in the world at large. End-to-End (E2E) encryption in mobile communication applications delivers confidentiality between users, defending messages against snooping. Several widespread communication tools (WhatsApp, Signal, Telegram, iMessage) have implemented end-to-end encryption, as a major selling point. Yet, the understand of the security goals (confidentiality, integrity, authentication) remain vague to users, such as how the security goals offer protection, and if they value that protection. In this research, we conducted a performance evaluation analysis on various cryptographic algorithms such as Advance Encryption Standard (AES), Blowfish, Data Encryption Standard (DES), Hybrid (AES-RSA) and Triple Data Encryption Standard (3DES). To ascertain which would be better suited for End-To-End Encryption security in a network or system. The following performance evaluation parameters such as encryption/decryption time, different key size, CPU utilization, memory consumption rate and overall process time, was utilized. Through the performance evaluation analysis of various cryptographic algorithms, it was concluded that the hybrid (AES-RSA) encryption algorithm is more secured. And when applied in End-To-End Encryption security scenario, can enhance the encryption effectiveness, key organization and security; thus, eliminating the gaps inherent in AES. This would serve as a guide to understanding how End-to-End Encryption works in meeting the security goals.*

Key Words: Cryptography, End-to-End Encryption, Security, Symmetric, Asymmetric, Hybrid

1. INTRODUCTION

1.1 Background of the Study

With the growing use of the internet as a medium for communication, it becomes imperative to secure personal and business' online communications. Consequently, the motivation to execute attacks increases and preventing against these attacks using technology that is almost unbreakable was considered. This technology, if employed accurately, could avert large-scale attacks. End-To-End Encryption suggests a maintainable answer to the continuing challenges of internet security [1]. End-to-end encryption describes the process of secure exchange of data from sender to recipient; preventing third-parties from accessing the data during transmission. All information is encrypted by the sender and the recipient decrypts it. During transmission, the content is completely encrypted, which means that no third parties can access or tamper with it during transmission [2]. Several cryptographic algorithms are used alone or combined for the encryption purposes.

The components of End-To-End Encryption includes [1]: The **identity** component authenticates users. The **protocols** component handles the key exchange and the algorithm. The **algorithm** uses scientific process to encrypt the data, and it cannot be decrypted without the predetermined key. Secure **implementation and operation** ensure the End-To-End Encryption process is not vulnerable to attacks on the hardware side. These components work together to deliver a system that operates efficiently to offer the best security to end users.

1.2 Rationale for the Study

Internet security is an extremely vital issue in computer science, due to the increasing acceptance of online communication. Since e-mailing services became public, questions arose about how secure they are. Furthermore, the need to secure the internet was made popular by shopping, banking, and other financial transactions via the internet [1]. Is End-To-End Encryption the best technology to ensuring data security? In recent times, most of the communications between clients and servers are secured using Transport Layer Security. However, communications between clients are not yet secured [3]. With plaintext communication vulnerable to hackers [4]. Consequently, security researchers have proffered the usage of End-To-End Encryption.

1.3 Scope of the Study

This research focused on providing a performance evaluation experiment of various cryptographic algorithms, such as Advance Encryption Standard (AES), Blowfish, Data Encryption Standard (DES), Hybrid (AES-RSA) and Triple Data Encryption Standard (3DES), so as to ascertain which is best suited for End-To-End Encryption security. This research would be limited to following performance evaluation parameters such as encryption/decryption time, different key size, CPU utilization, memory consumption rate and Overall Process Time.

1.4 Justification of the Study

An information's value is proportional to the information risk, thus, if the information is high valued then there is a great need for protection and security. The various cryptographic algorithm has their own advantages and

disadvantages. With the aim of apply a suitable cryptographic algorithm to the End-To-End Encryption security, data concerning performance, strength and weakness of each use case cryptographic algorithms is essential.

1.5 Significance of Study

This research aims to show that various cryptographic algorithms have their advantages and disadvantages. And though might be able to provide the needed security, still fail under certain conditions. In choosing the cryptographic algorithm to employ in End-To-End Encryption, this research can serve as a guide to students, organizations and other researchers, to making the right choice.

2. LITERATURE REVIEW

2.1 Cryptography

There has always been an inherent need for Human being to interact and share information privately and publicly. This gave rise to the science of encoding, such that messages are scrambled in a way that only authorized individuals can access it. Cryptography is defined as the science of obscuring the communications, introducing confidentiality in data security. On the other hand, cryptography is the study of designing or producing the secret message, that is, ciphertext of the original communication for protected transmission between sender and recipient [5].

2.2 Categories of Cryptographic Algorithm

There are numerous cryptographic algorithms in extensive use and are categorized as follows:

2.2.1 Symmetric Encryption Algorithms

In symmetric encryption algorithm, a single encryption key is used in the encryption and decryption process. The encryption key is conveyed to the sender and recipient before the encryption/decryption processes. So, the encryption key is vital and its strength is contingent on its length (in bits). Symmetric encryption algorithms examples are RC2, RC5, Advance Encryption Standard (AES), Blowfish, Data Encryption Standard (DES), Hybrid (AES-RSA) and Triple Data Encryption Standard (3DES) [5].

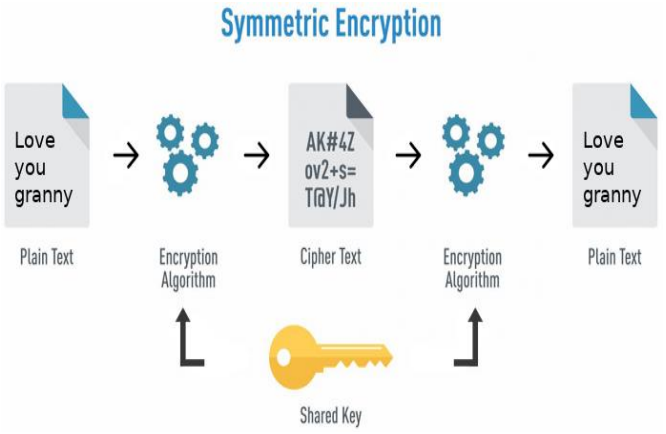

Fig -1: Symmetric Encryption Process

1. Data Encryption Standard (DES): This is a Feistel-type Substitution-Permutation Network (SPN) 64-bit block cipher with 56-bit key. A 16 rounds system with an overall 56-bit key permuted into 16 48-bit subkeys, for each round. For decryption, the order of subkeys is reversed. It is susceptible to brute-force attacks [6].

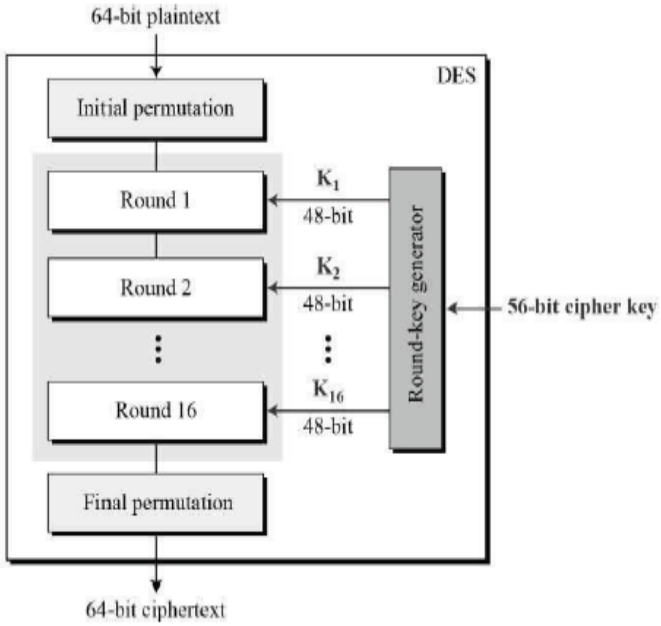

Fig -2: Structure of Data Encryption Standard

2. Triple Data Encryption Standard (3DES): This is a process in through which data is encrypted using 56 bit two keys. Its process follows EDE model, which says data must be sequentially encrypted twice and decrypted once. First, it encrypts using one encryption key, then decrypts using a different encryption key, and finally encrypts using same encryption key. For encryption, EDE uses only 168-bit out of 192-bits keys. Even though we do not use the last 8-bits; it is still secure [7].

3. Advanced Encryption Standard (AES): It entails several rounds; each performing a number of transformations, using

a round key produced from the encryption key. The number of rounds is contingent on the block count and encryption key length. Encryption/Decryption starts with a transformation, accompanied by a number of rounds, and finally ends with a round which is different. This makes decryption possible by reversing the encryption process. Each round has four transformations: AddRoundKey, SubBytes, ShiftRows and MixColumns. An encryption key length (size) of 128bits would require 10 rounds [8]. It has different key lengths such as 128bits, 192bits, and 256 bits.

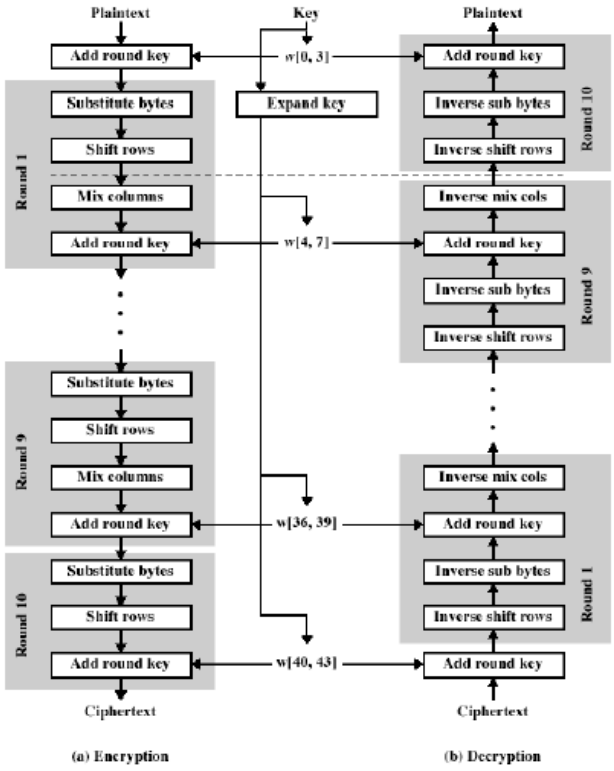

Fig -3: Structure of Advanced Encryption Standard

4. Blowfish: Blowfish is a symmetric block Feistel Network cipher encryption algorithm, with a 64-bits block length. The block size is 64 bits and iterates a simple encryption function 16 times. The key length is variable from 32 bits to 448 bits, making it perfect for information security. Padding are applied to messages that are not a multiple of 8 bytes in size [9].

2.2.2 Asymmetric Encryption Algorithms

In asymmetric encryption algorithms, two types of keys called Private keys and Public Keys, are utilized. The recipient's public key is used to produce a ciphertext from the plaintext. Then the ciphertext can only be decrypts using the recipient's private key [10]. The private key is known by the authorized person only. But the public key is stored in the public domain for ease of assess [11]. Asymmetric encryption algorithms examples are Digital Signatures, Rivest-Shamir-Adleman (RSA) and so on.

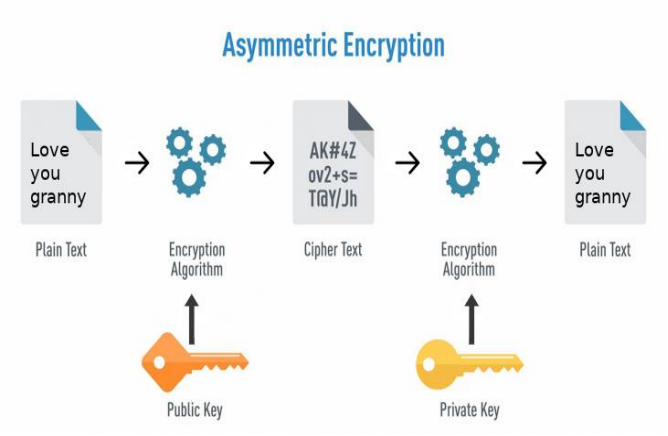

Fig -4: Asymmetric Encryption Process

1. Rivest-Shamir-Adleman (RSA) Algorithm: This is based on number theory, using two prime numbers or mathematical operation to randomly produce the public and private keys. The public key (which is public) is used for encryption, and the private key (which is private) is used for decryption. Sender encrypts the communication using public key of the recipient and when the communication is received, the recipient can decrypt it with its private key [12].

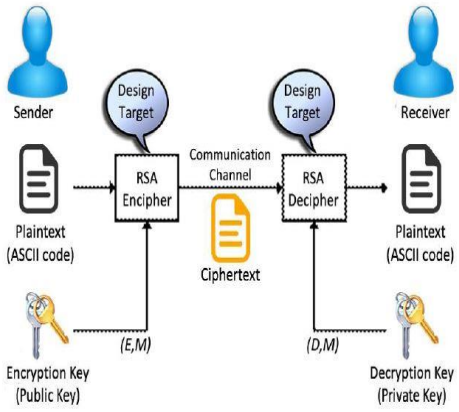

Fig -5: RSA Algorithm Structure

2.2.3 Hybrid Encryption Algorithm

Hybrid encryption is a blend of symmetric and asymmetric encryption methods. As well as the blend of two or more symmetric encryption methods. Symmetric algorithms are used for encryption of messages rather than asymmetric. Thus, the asymmetric algorithm is used for the purpose of safeguarding and protecting the data. The secret key which solves the problem of key exchange can be sent securely. This blend of symmetric and asymmetric encryption benefits from the strengths of each encryption type. For example, Hybrid (AES-RSA) Encryption Algorithm can take benefits of Advanced-Encryption-Standard (AES) algorithm and Rivest-Shamir-Adleman (RSA) algorithm, making it difficult to compromise its security.

2.3 Review of Related Literatures

Ermoshina, Musiani, and Halpin [13] aimed at giving an overview of the different core protocols used for decentralized chat and email-oriented services. This work is part of a survey of 30 projects focused on decentralized and/or end-to-end encrypted internet messaging. Results show open standards for encrypted e-mail and chat are still not seeing widespread use, and a new generation of end-to-end encrypted messaging protocols offering better security properties are rapidly gaining traction. Although most are not yet standardized or decentralized.

There are many variants of end-to-end encryption schemes for different communication patterns. Nabeel and Doha [14] systematically analyzed the security of these different variants against three types of passive adversaries and one type of active adversaries. Results show that the security of some of these systems are broken under these threat models and what can be done to ensure confidentiality in such systems. Most of the end-to-end encrypted systems are secure against only the weakest passive adversaries. Further, these systems are broken not by cryptanalysis of underlying cryptographic algorithms but by flawed system designs and security assumptions. Identify that unencrypted metadata and access patterns make these systems susceptible to inference attacks.

Information Systems security cannot be fully apprehended if the user lacks the required knowledge and skills to effectively apply the safeguard measures. Hameed, Asanka, and Arachchilage [15] carried out a study using the method of Systematic Literature Review using 42 extant studies to evaluate individual self-efficacy for Information Systems security innovation adoption. The results highlighted that individual self-efficacy is a significant attribute of Information System security innovation adoption. 92% of the studies found self-efficacy as significant attribute in Information System security innovation adoption. Also, approximately 71% of the studies verified the association between self-efficacy and Information System security adoption as moderate significance or strong significance. The major limitation of this analysis was the inadequacy of studies that examined individual self- efficacy on Information System security innovation adoption. The result would be more accurate and better explained if analyzed with more studies.

Research indicates that many users have difficulty using End-to-End Encryption (E2EE) tools correctly and confidently, as well as recognizing their security benefits, in part because of incorrect mental models. Bai, Pearson, Kelley, and Mazurek [16] took the first step toward providing high-level, roughly correct information about end-to-end encryption to non-experts. In a lab study, participants (n=25) were asked about their understanding of E2EE before and after a tutorial we created, as well as which information they found most useful and surprising. Participants' understanding of the benefits and limitations of E2EE improved. They found information about confidentiality, risks and weaknesses most useful, surprising, and compelling to pass on to others. Some confusion about integrity and authenticity remained.

Messages sent on social media can be seen or used and obtained by others. Given the current many types of text messages that must not be known by others, or secret messages. Because social media is free, users cannot request security facilities for their messages. Tarigan, Sunandar, Sinuraya, Matondang, and Ginting [17] proposed a technique using the triangle chain cipher algorithm; one of the encryption algorithms that operates based on classical encryption (cryptography), especially in character substitution techniques. The solution is able to secure communication but still require further testing.

3. METHODOLOGY

This session presents comprehensive explanation of the methodology used in achieving this research work. The work proposes to ascertain which cryptographic algorithm is best suited for End-To-End Encryption security. By carrying out a performance evaluation analysis on various cryptographic algorithms; such as 3DES, DES, AES, Blowfish and Hybrid (AES-RSA). The following performance evaluation parameters such as encryption time, different key size, CPU utilization, memory consumption rate and overall process time, would be utilized.

3.1 Simulation Parameters

For easy understanding of the various stages involved, the simulation process is divided into three modules. The entire simulation process is applied to all use case algorithms respectively.

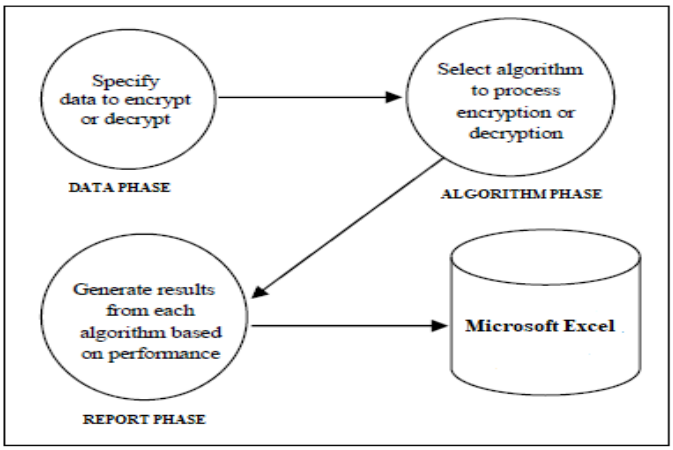

Fig -6: Outline of the Simulation Process

1. **Data Phase:** This is the data entry phase. The data can take various forms such as numeric (1234, 666889.453), alpha-numeric (A45FBE3), characters (this is my

"project"), images (jpg, jpeg), audio (mp3, wav, m4a), video (mp4, HD).

2. **Algorithm Phase:** In this phase, the use case cryptographic algorithms utilized in encrypting and decrypting the data is selected. The data entered is encrypted and decrypted multiple times with each use case cryptographic algorithm (five times in all).
3. **Report Phase:** In this phase, the results are displayed. Once the data is entered and treated (encrypted and decrypted), the result is displayed. The report will include the results of the use case cryptographic algorithms based on the following parameters: encryption/decryption time, different key size, memory consumption, CPU utilization and overall process time. The ensuing report will be recorded on Microsoft Excel. This result is crucial to defining the most secure and suitable cryptographic algorithm for use in end-to-end encryption security scenario.

3.2 Performance Evaluation Parameters

In this research, the performance evaluation parameters used to analysis comparatively the use case cryptographic algorithms below [18]:

1. **Encryption and Decryption Time:** It is imperative that the cryptographic algorithms are fast enough to meet real-time necessities. Consequently, the encryption/decryption time will be determined by recording the **Encryption Time** (time taken to transform plaintext to ciphertext) and **Decryption Time** (time taken to transform ciphertext to plaintext).
2. **Different Key Length (Size):** In the encryption process, key management is a vital feature, showing how the data is transformed using a variable key length which is longer. Each use case cryptographic algorithm utilizes a unique key length which is used in the encryption/decryption process. The length of the key is proportional to its security strength.
3. **CPU Utilization Time Period:** Cryptographic algorithms utilize significant system resources such as memory, CPU. Thus, the various resources utilized by each use case cryptographic algorithm will be recorded and evaluated to ascertain which algorithm utilizes more system resources and how much more system resource was utilized.
4. **Memory Consumption Rate:** Cryptographic algorithm requires different memory size for its operations. This is contingent on the number of operations, key size utilized, initialization vectors and operations type. It is necessary the memory required is as small as possible.
5. **Overall Process Time:** This is the time taken to achieve the overall system process for each use case algorithm. Where Start time = Tx_1 (ms) and Finish time = Tx_2 (ms); time taken for the whole process is given in equation (1).

Overall process time (Tx) = $Tx_2 - Tx_1$ ..... (1)

3.3 Simulation Implementation

The experiment was carried out using CORE i7 64bit processor with 8GB of RAM. PyCharm IDE for windows applications was utilized to compile the simulation using the interpreter settings. And also, to implement the algorithms in python programing language. The packages utilized are: The package **cryptography**, a cryptographic standard library, provides cryptographic recipes and primitives. **PyCryptodome** is a self-reliant python package of low-level cryptographic primitives that has been enhanced to add more implementations and fixes. **Psutil** is a cross-platform library for recovering data on processes that are running and other system utilization, such as CPU, memory, disks, network, sensors.

The sizes of the input files used are shown in Table 1 below and consists of text, images and audio. The output of the encryption process for each file is in turn used as input for decryption. The study utilized the same input files through the experimentation, to ensure the same memory and processor conditions for all use case cryptographic algorithms. The experimentation will be carried out a multiple time to guarantee results are consistent and valid. Also, cryptographic algorithms utilize a considerable volume of system resources (CPU time, memory and computation time).

Table -1: Data Table

File name	File sizes
Text	400, 1500, 2048, 3000 (bytes)
Image	400, 1500, 2048, 3000 (kb)
Audio	26.1(Mb)

4. RESULTS OF EXPERIMENT

This section displays the results from executing the simulation parameters on the PyCharm IDE. The text and image file were used to ascertain the encryption and decryption time. The results illustrate the effect of varying data sizes and the effect of encryption/decryption mode for each use case cryptographic algorithms. While the audio file was used to ascertain the memory rate, CPU utilization, and overall process time. The overall result is transferred to Microsoft excel for more investigation and graphs plotted in Chart 1 – Chart 8 for each resourced measured. These results of each use case algorithms were conducted multiple times and average calculated to get the final value.

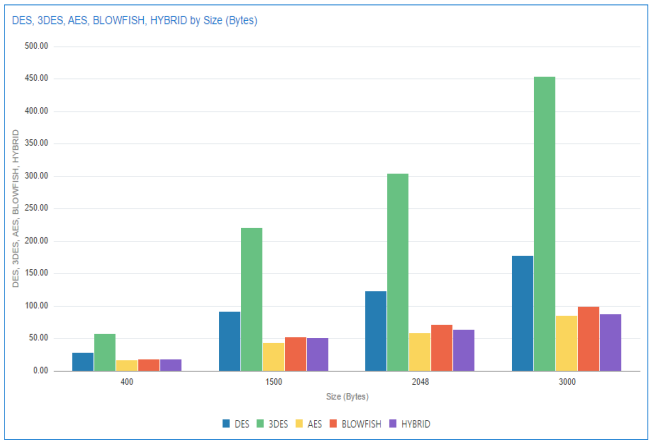

Chart -1: Encryption time for Text

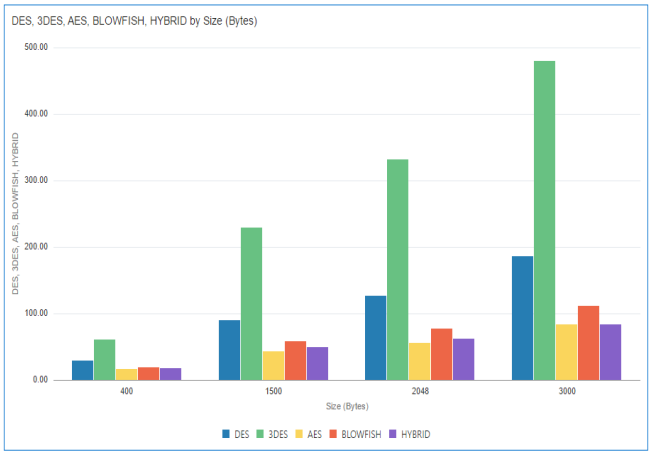

Chart -2: Decryption time for Text

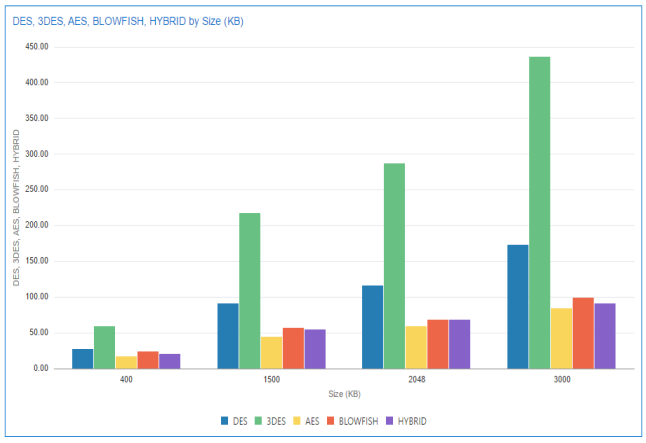

Chart -3: Encryption time for Images

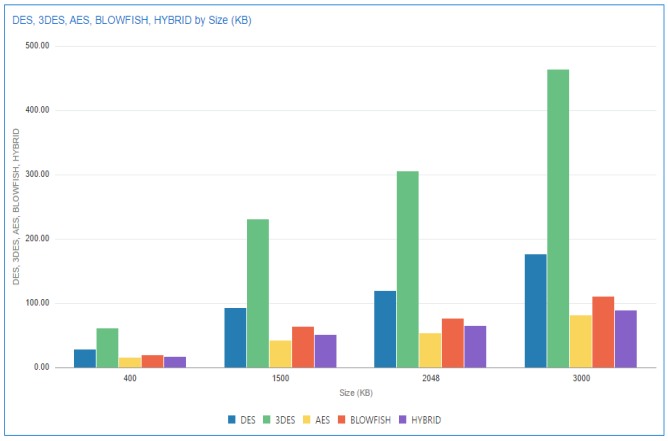

Chart -4: Decryption time for Images

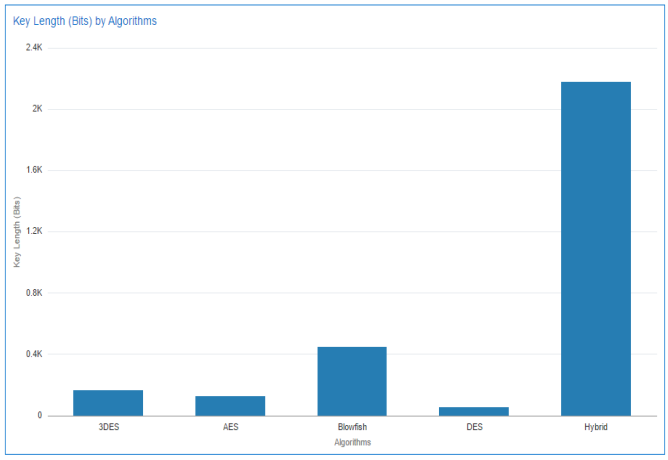

Chart -5: Key length of the use case cryptographic algorithms

Chart 5 illustrates that Hybrid displays the highest key length, followed by Blowfish. While DES displays the least key length. The key length tells us the strength of encryption in relations to the difficulty of discovering the key. Such that longer key length provides stronger encryption. Hybrid combines the encryption keys of AES and RSA, leading to high security of information.

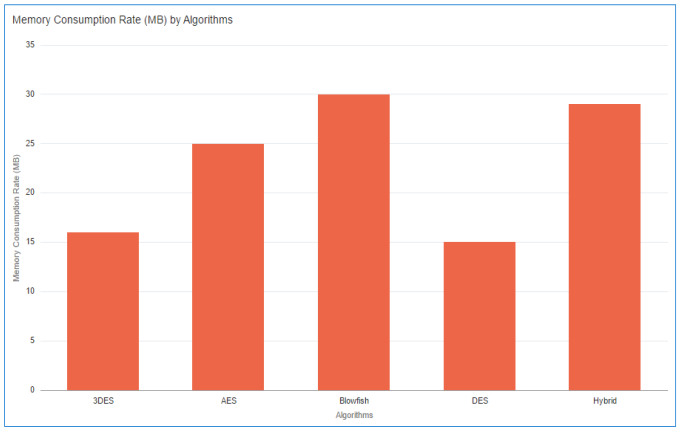

Chart -6: Memory Consumption Rate of the use case cryptographic algorithms

Chart 6 illustrates that Blowfish consumes the second highest memory. It utilizes a key-dependent lookup tables; thus, performance is contingent on memory and caches handling. Hybrid consumes the highest memory for encryption and decryption. It combines the cascading instance of AES and RSA with distinct keys. 3DES recycle DES operation by cascading three occurrences of DES with different keys. DES take least memory.

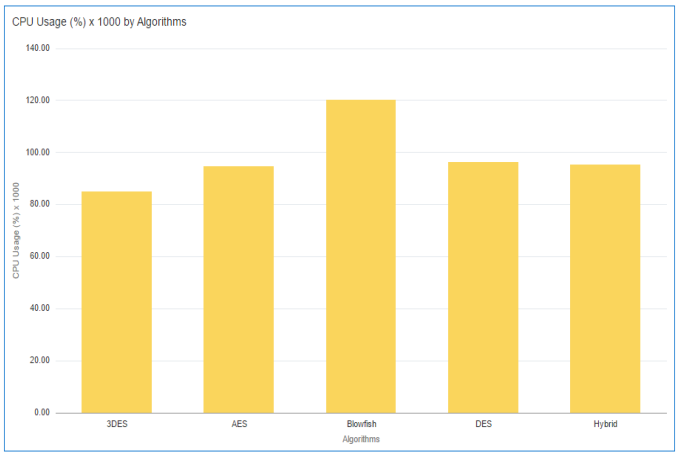

Chart -7: CPU Utilization Time of the use case cryptographic algorithms

Chart 7 illustrates that Blowfish utilizes the highest CPU utilization time for encryption and decryption. This is closely followed by DES then Hybrid then AES in CPU utilization time. While 3DES uses the least CPU utilization time.

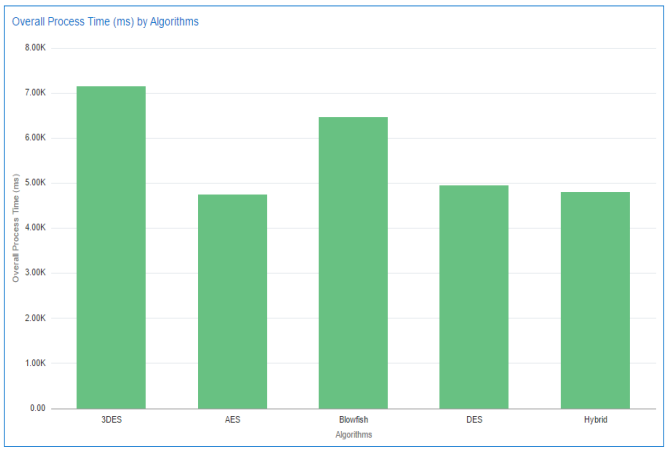

Chart -8: Overall Process Time of the use case cryptographic algorithms

Chart 8 shows that AES has the least overall process time, closely followed by hybrid. While 3DES displays the highest overall process time, followed by Blowfish. Overall process time tells us the total time elapsed to move a unit of work from beginning to end of the process. This shows that AES has a best overall process time than other algorithms.

5. DISCUSSION OF FINDINGS

From the experimental results above, advanced encryption standard (AES) performed best in comparison to other algorithms. Followed closely by hybrid (AES-RSA) then Data Encryption Standard (DES) respectively. But in terms of security, hybrid encryption is more secure combining the speed benefit of AES algorithm and security and key management benefit of RSA algorithm. It is noteworthy that RSA algorithm is not used for encrypting the data directly, because the encryption and decryption time is not fixed. Such that, if the file is larger, the encryption time will be longer. Considering the efficiency of AES algorithm, it is used for the encryption of the data initially producing a cyphertext of fixed-length, and then RSA algorithm is used for the encryption of the ciphertext and the AES encryption key. This will critically enhance and ensure the operational efficacy and security. The decryption time recorded for the use case cryptographic algorithms increases with the varying data size, with AES algorithm having small increase. For Triple Data Encryption Standard (3DES), the decryption difficulty increased, due to the triple-layered encryption process. The aim of this comparative analysis is to determine the appropriate cryptographic algorithm suitable for the End-To-End Encryption security. How secure the End-To-End Encryption is contingent on how long it takes to compromise the encryption and how high the cost implication will be. To achieve this, two end-to-end encryption scenarios using AES and Hybrid will be considered respectively:

Scenario 1: Advanced Encryption Standard (AES) End-To-End Encryption

In the AES end-to-end encryption scenario, the AES algorithm uses 128-bit key length for its encryption and decryption processes. From the comparison, AES algorithm has an exceptional performance and a relatively fast encryption speed. Its encryption efficiency is high and suitable for encryption/decryption of data of large sizes. Considering the security concerned, the AES algorithm has a relatively high security than DES and 3DES algorithms. For resource consumption, AES algorithm consumes less compared to Blowfish and Hybrid algorithms.

In the encryption process, the encryption key is used to encrypt the source file (plaintext) to generate a ciphertext.

Fig -7: AES Encryption Process

In the decryption process, the encryption key is used to decrypt the ciphertext to generate the source file (plaintext).

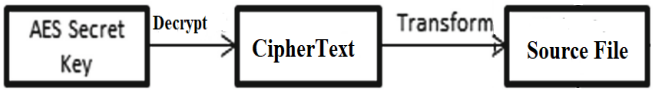

Fig -8: AES Decryption Process

This works better when applied to a network or system that requires unidirectional communication and does not require the distribution of the secret key. However, when applied in a network or system that requires bidirectional communication, AES algorithm has a drawback in key management. Thus, making the secure distribution and management of the encryption key difficult. This makes it possible under certain conditions for AES algorithm to be hacked, such as:

1. Since same key is used in the end-to-end encryption/decryption process, it is essential to decide on the encryption key earlier to ensure secure distribution, lest the information be cracked;
2. Each time the system is used; a unique key that is not known to others is used. This increases the number of encryption keys; thus, causing a management bottleneck.

The summary of AES End-To-End Encryption security goals are as follows:

1. **Confidentiality:** AES will be used to achieve this security goal on high level.
2. **Integrity:** AES does not have this objective achieved. As the secret key can be compromised when distributed in a bidirectional communication.
3. **Authentication:** In a unidirectional communication, AES have this objective achieved. However, in a bidirectional communication, it fails due to the fact that the secret key can be compromised when distributed.

Scenario 2: Hybrid (AES-RSA) End-To-End Encryption

Owing to the drawback inherent in AES End-To-End Encryption, a Hybrid End-To-End Encryption was considered. Hybrid encryption combines the benefits of two cryptographic algorithms: the encryption speed benefit of AES algorithm and security and key management benefit of RSA algorithm. Consequently, combining the strength of both cryptographic algorithms to secure communications. This can be applied to network or system that requires uni- and bi- directional communication.

In the encryption process, the AES encryption key is initially employed to secure the source file (plaintext) to produce the first ciphertext (ciphertext 1), the RSA public key is then employed to encrypt the first ciphertext (ciphertext 1) and AES encryption key to produce the second ciphertext (ciphertext 2). Such that, even if the information (ciphertext 2) is compromised, the data encrypted using RSA public key can only be accessed using the matching RSA private key.

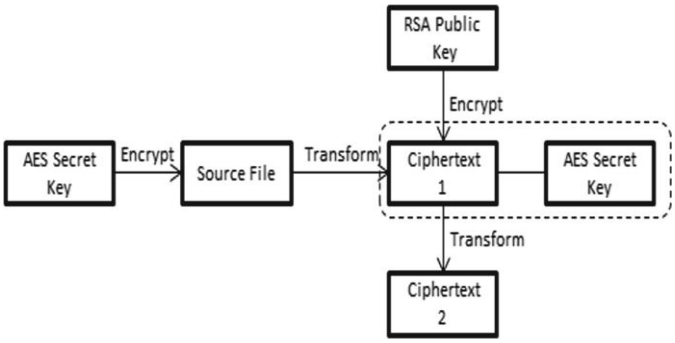

Fig -9: Hybrid (AES-RSA) Encryption Process

The decryption process is opposite of the encryption process. The RSA private key is utilized to decrypt the second ciphertext (ciphertext 2) to get the first ciphertext (ciphertext 1) and AES encryption key, and finally the first ciphertext (ciphertext 1) is decrypted by the AES encryption key to get the source file (plaintext).

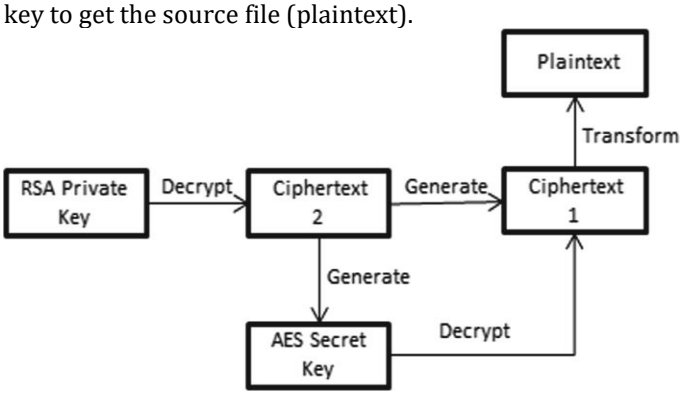

Fig -10: Hybrid (AES-RSA) Decryption Process

The key management feature of hybrid encryption includes:

1. **Key generator**: The following keys have to be available to fulfill the whole process of hybrid technique, AES encryption key, RSA public key, RSA private key.
2. **Key storage:** Public keys are published in place on high availability. Private keys are kept in secured place and there is no need to get them known by third-party.
3. **Key transmission:** The AES-secret key is encrypted and transmitted using RSA public.

The benefits of this process:

1. No sharing encryption key is required. Only the recipient's public key is needed.
2. A unique key is used to encrypt each data.
3. The fast AES algorithm encrypted and decrypted the large data; only the encryption key and ciphertext, uses the slower RSA algorithm.
4. If there is a compromise, the data is still fully protected.
5. If multiple versions of the same data need to be sent; multiple copies of ciphertext 1 and the AES encryption key will be stored, and for each recipient use their public key to encrypt ciphertext 1 and AES encryption key before sending.

The summary of Hybrid End-To-End Encryption security goals are as follows:

1. **Confidentiality:** AES will be used to achieve this security goal on high level. Using AES, symmetric cryptography algorithm, is better since it handles bigger data size than RSA.
2. **Integrity:** AES and RSA together will assist to have this objective achieved. The computation happens inside the ciphertext and the verification happen on reception stage. Considering the computing time is an important aspect which has to be optimized as much as possible.
3. **Authentication:** The RSA public key can be used to achieve authentication. This is because it has an associated private key which no one has access to. Thus, allowing positive and unique identification.

6. CONCLUSION AND RECOMMENDATION

This research proposes to ascertain which cryptographic algorithm is best suited for End-To-End Encryption security. Through the performance evaluation analysis of various cryptographic algorithms, it was concluded that hybrid encryption algorithm is more secured. And when applied in End-To-End Encryption security scenario, can enhance the encryption effectiveness, key organization and security; thus, eliminating the gaps inherent in AES. This was built based on data encryption and key encryption features, thus, providing a higher level of security and efficient. The security of data in this hybrid is achieved by AES-128 algorithm to confirm the confidentiality. Integrity was ensured by using RSA to encrypt AES secret key. Authentication was achieved using RSA public key. This approach offers a resolution to various security deficiencies. The hybrid encryption can be design for system and software application were effective data protection is needed.

The following area are recommended for further research:

1. To study how hybrid encryption algorithms can be employed in one algorithm.
2. To analyze how unique algorithm can accommodate high processing time and eradicate key sharing.
3. The evaluation of the Hybrid End-To-End Encryption security in an application environment.

REFERENCES

[1] E. Wehner and E. Moran, "END TO END ENCRYPTION: AN ANSWER TO SECURITY CONCERNS IN THE PRIVATE SECTOR," University of Pittsburgh Swanson School of Engineering, pp. 1-8, 2017.

[2] R. Margaret, "End-To-End Encryption (E2EE)," 31 July 2015. [Online]. Available: https://www.google.com/amp/s/searchsecurity.techtarget.com/definition/end-to-end-encryption-E2EE%3famp=1.

[3] N. Unger, S. Dechand, J. Bonneau, S. Fahl, H. Perl, I. Goldberg and M. Smith, "SoK: Secure Messaging," in IEEE Symposium on Security and Privacy, 2015.

[4] R. Abu-Salma, M. A. Sasse, J. Bonneau, A. Danilova, A. Naiakshina and M. Smith, "Obstacles to the Adoption of Secure Communication Tools," pp. 1-17, 2017.

[5] M. Akanksha, "A Research paper: An ASCII value based data encryption algorithm and its comparison with other symmetric data encryption algorithms," Computer Science. & Engineering. JIET Group of Institutions, vol. 4, no. 9, 2012.

[6] H. O. Alanazi, B. B. Zaidan, A. A. Zaidan, H. A. Jalab, M. Shabbir and Y. Al-Nabhani, "New Comparative Study Between DES, 3DES and AES within Nine Factors," JOURNAL OF COMPUTING, vol. 2, no. 3, pp. 152-157, 2010.

[7] R. Anup and R. Suchithra, "Image Encryption using Triple DES Algorithm," Imperial Journal of Interdisciplinary Research (IJIR), pp. 969-974, 2017.

[8] P. Jindal and B. Singh, "STUDY AND PERFORMANCE EVALUATION OF SECURITY-THROUGHPUT TRADEOFF WITH LINK ADAPTIVE ENCRYPTION SCHEME," Department of Electronics and Communication Engineering, pp. 1-14, 2012.

[9] S. Potteti and N. Parati, "SECURED DATA TRANSFER FOR CLOUD USING BLOWFISH," International Journal of Advances In Computer Science and Cloud Computing, vol. 3, no. 2, pp. 17-22, 2015.

[10] S. Gurjeevan and K. S. Ashwani, "Throughput Analysis of Various Encryption Algorithms," International Journal of Computer Science and Technology, vol. 2, no. 3, September 2011.

[11] S. A. E. Diaa and M. A. K. Hatem, "Evaluating the Performance of Symmetric Encryption Algorithms," International Journal of Network Security, vol. 10, no. 3, pp. 216-222, May 2010.

[12] K. Aman, J. Sudesh and M. Sunil, "Comparative Analysis between DES and RSA Algorithm's," International Journal of Advanced Research in Computer Science and Software Engineering, vol. 2, no. 7, pp. 386-391, July 2012.

[13] K. Ermoshina, F. Musiani and H. Halpin, "End-to-End Encrypted Messaging Protocols: An Overview," INSCI 2016, LNCS 9934, p. 244–254, 2016.

[14] M. Nabeel and Q. Doha, "The Many Faces of End-to-End Encryption and Their Security Analysis," in 2017 IEEE 1st International Conference on Edge Computing, 2017.

[15] M. A. Hameed, N. Asanka and G. Arachchilage, "Understanding the influence of Individual's Self-efficacy for Information Systems Security Innovation Adoption: A Systematic Literature Review," pp. 1-19, 2018.

[16] W. Bai, M. Pearson, P. G. Kelley and M. L. Mazurek, "Improving Non-Experts' Understanding of End-to-End Encryption: An Exploratory Study," University of Maryland, pp. 1-15, 2020.

[17] P. Tarigan, H. Sunandar, B. Sinuraya, Z. A. Matondang and G. Ginting, "Implementation Of Triangle Chain Cipher Algorithm in Security Message of Social Media," Journal of Physics, pp. 1-10, 2020.

[18] J. B. Awotunde, A. O. Ameen, I. D. Oladipo, A. R. Tomori and M. Abdulraheem, "Evaluation of Four Encryption Algorithms for Viability, Reliability and Performance Estimation," NIGERIAN JOURNAL OF TECHNOLOGICAL DEVELOPMENT, vol. 13, no. 2, pp. 74-82, 2016.