NIPES

Journal of Science and Technology Research

Journal homepage: www.nipesjournals.org.ng

A Comparative Analysis of Symmetric Cryptographic Algorithm as a Data Security Tool: A Survey

Ubochi Chibueze Nwamouh[1*], Bashir Olaniyi Sadiq[1], Kelechi Ukagwu John[1], Stephen Nnamchi Ndubuisi[2]

[1]Department of Electrical, Telecommunication and Computer Engineering, Kampala International University, Uganda
[2]Department of Mechanical Engineering, Kampala International University, Uganda
*Corresponding author: ubochi@kiu.ac.ug

Article Info

Keywords: *cryptographic algorithm, data security, DES, 3DES, AES, encryption, decryption*

Received 13 June 2023
Revised 09 July 2023
Accepted 09 August 2023
Available online 27 August 2023

https://doi.org/10.5281/zenodo.8313097

Abstract

Network security is becoming a significant and difficult subject that is growing quickly. Attacks and risks to information and internet security are becoming harder to identify. As a result, encryption has been developed as a remedy and is crucial to information security systems. To safeguard the shared data, several strategies are required. In this study, we examined and contrasted the performance of three cryptographic algorithms: DES, 3DES, and AES as applicable to e-voting system. The symmetric encryption techniques mentioned above have been compared. The amount of CPU time, memory, and battery power used by these methods is substantial. A thorough comparison of the efficacy of each algorithm is provided. The criteria used for comparison include speed, block size, key size, etc. Compared to alternative DES and 3DES algorithms, AES performs better.

1. Introduction

Information and communication technology (ICT) devices have accelerated the development of data processing operations by transferring the information that our society has long relied on as a tool for decision-making [1]. In the era of the Industrial Revolution 4.0 and Society 5.0, data serves many important purposes. Data security, however, may take several forms, depending on the situation and need. Regardless of whoever is involved, each participant in a transaction must have some level of assurance that certain objectives relating to data security have been met. In Table 1, some of these objectives are stated.

Table 1: Some data security objectives.

Security objectives	Description
privacy or confidentiality	keeping information private and only allowing authorized people to access it.
data integrity	guaranteeing that data hasn't been tampered with or manipulated in any way.
entity authentication or identification	confirmation of a person's or thing's identification (e.g., a person, a computer terminal, a credit card, etc.).
message authentication	Authenticating the data origin; also known as corroboration of the information source.
Signature	a way to connect data to an object.
authorization	transfer of formal approval to perform or be something to another organization
Validation	a way of timely granting permission to utilize or alter resources or information.

access control	limiting privileged entities' access to resources
certification	information that has been approved by a reliable source.
timestamping	keeping track of when information was created or first existed.
Witnessing	confirming that information was produced or exists by a source other than the original originator.
Receipt	acknowledgment of the receipt of information.
confirmation	recognition of the provision of services.
Ownership	a way of giving a company the authority to utilize or transfer a resource to others
Anonymity	a process's identity of a participant being hidden.
non-repudiation	prohibiting the renunciation of prior obligations or deeds.
Revocation	retraction of permission or certification.

Today, data security is a crucial concern. Data interruption, interception, manipulation, and fabrication are some of the threats to data security. The ease with which data may be exchanged across several technology platforms pose a severe danger to data security. Therefore, one of the most difficult jobs in recent years has been ensuring the integrity and security of data [2].
The previous sentence suggests that in the new technological invasion we are currently experiencing, security is no longer an option. Today, one of the most important topics for both industrial and research organizations to focus on is data security.
Networks were introduced to speed up data transport, but this generated security concerns. The discipline of data security and cryptography was created as a result of the problems associated with utilizing the Internet for data transport [1]. Due to e-initiatives, data security and integrity are now under threat (such as e-voting, e-Commerce, e-banking etc.). Because there are always hackers whose goal is to steal data, these improvements have increased user awareness of security risks. As a consequence, the frequency and complexity of security events and specific vulnerabilities have risen, as depicted in Figure 1.

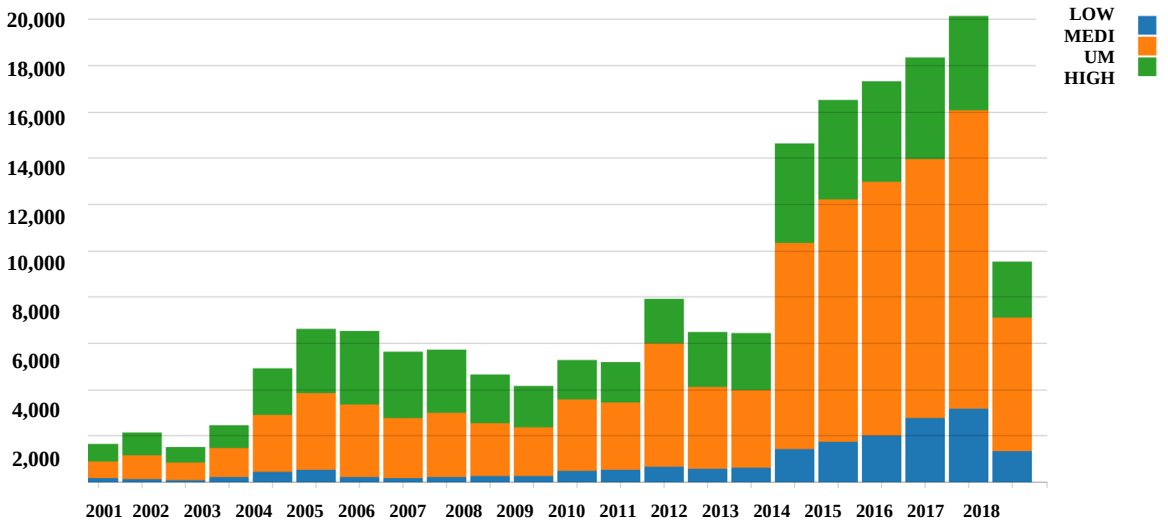

Figure 1: The evolution of data threat severity [3]

Due to on-going advancements in Internet technology, the security problem has grown so crucial that it only becomes worse over time [4]. According to Figure 1 above, the amount of threats to electronic data reported to the National Vulnerability Database (NVD) has been steadily (no fluctuations), gradually (sometimes fluctuating), exponentially in 2017, and then constantly each year afterward [3]. [3] speculate that the surge in 2017 may have been caused by an increase in the use of electronic data from software packages that are included in NVD. However, it's crucial to

keep in mind that, as the graph demonstrates; electronic data exploitation has predominated over the decade as seen in Figure 2.

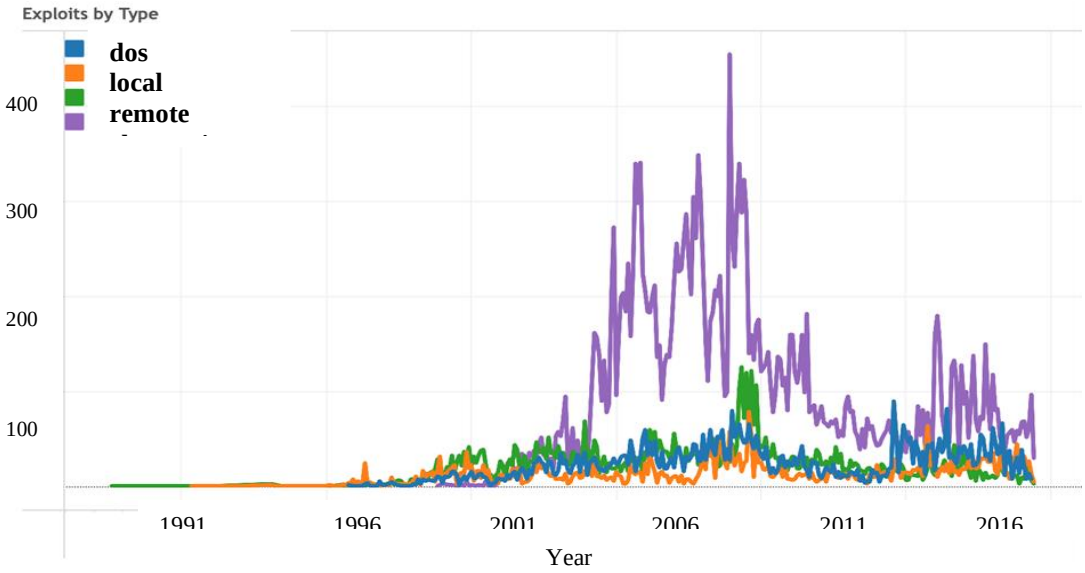

Figure 2: Data exploitation by kind over ten years [3]

All types of data (including videos, photographs, and messages) are transferred between users over an insecure channel since data is being carried across it and stored on servers and clouds. Therefore, it is easier for hackers or eavesdroppers to collect and decode electronic data. Finding new security solutions that can manage this massive degree of change in data exchange is more important than ever. But the rapid development of networking technologies has fundamentally changed the way that data interchange has traditionally been done. Due to the volume of data that is exchanged over the internet, data security has become a difficult problem. When sent via an unsafe link, such information has to be safeguarded [5]. This necessitates the protection of data against disclosure, assurance of data and message correctness, and defense of systems against network-based intrusions [6]. This is when cryptography may be useful. One area of computer engineering that uses the approach of sending sensitive data through public networks is cryptography [7]. It is the process of converting plaintext data that is at rest or in transit into cipher text, which is a jumbled or incomprehensible format, as seen in Figure 3.

It has to do with the study of mathematical equations, codes, and rule-based computation (algorithms) for the parts of information security like secrecy, data integrity, and data authentication. Cryptography may be used to provide security in data display and secure the integrity and confidentiality of the data. It is a crucial tactic for protecting data from hackers and snoopers and is computationally secure [8]. By utilizing a cryptographic key to encrypt and decode plaintext as illustrated in equations 1 and 2, one may strengthen the security of cryptographic services such as confidentiality, approval, integrity, and accessibility of information.

The mathematical representation of the encryption process is as follows:

$$C=E(K,P) \quad (1)$$

Equation 1 is the mathematical equation for the encryption process. Where K stands for the key, P for plain text, E() for the encryption algorithm, and C for cipher text. The encryption function or algorithm E(.) produces C when used on P. On the other side, the equation 2 describes the decryption process.

$$P=D(K,C)=E(K,P) \quad (2)$$

D(), is the decryption algorithm that operates on C to produce P.

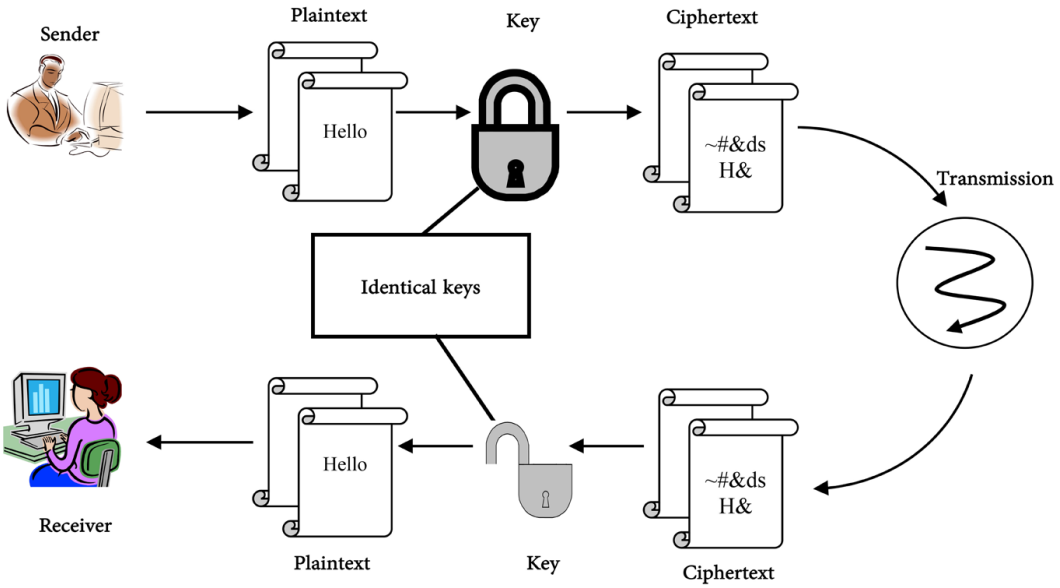

Figure 3: Cryptography as a concept [9]

Strong and lengthy cryptographic keys, which are essential elements in the encryption and decoding processes, enable secure encryption. Both symmetric and asymmetric key cryptography fall under the umbrella of cryptography. Asymmetric key cryptographies are not appropriate for large-sized documents, according to [9]. Asymmetric key cryptography is far slower in encrypting and decrypting data than symmetric key-based systems, and it also uses more CPU power [10]. Data Encryption Standards (DES), and Triple DES, are examples of common symmetric cryptographic methods [9]. Several of these techniques have previously been cryptanalysis even though symmetric encryption algorithms are faster and more effective than asymmetric encryption algorithms. Cryptanalytic attacks are categorized in Table 2. The in-depth search cryptanalysis has effectively attacked the DES and 3DES [11]. Figure 4 depicts the categorization of some common encryption techniques.

The Data Encryption Standard (DES) has been regarded as the encryption that is easiest to break and compromise throughout the last several decades. Cipher flaws were primarily responsible for making this feasible. A foundational 56-bit key was used to create DES. The F module, however, really utilized just 48 bits every round. However, a 64-bit data block size was chosen. As a consequence, assaults against encryption become simpler (when accounting for practically exponential increases in computer power) [12].

This paper's remaining sections are organized as follows. Section 2 provides a review of the literature on encryption techniques. In Section 3, popular block cipher-based encryption methods are completely explained. In Section 4, the AES cipher's efficiency and advantages are covered. Section 5 of the research comparison is described. The final section of the essay, Section 6, covers the conclusion of the research.

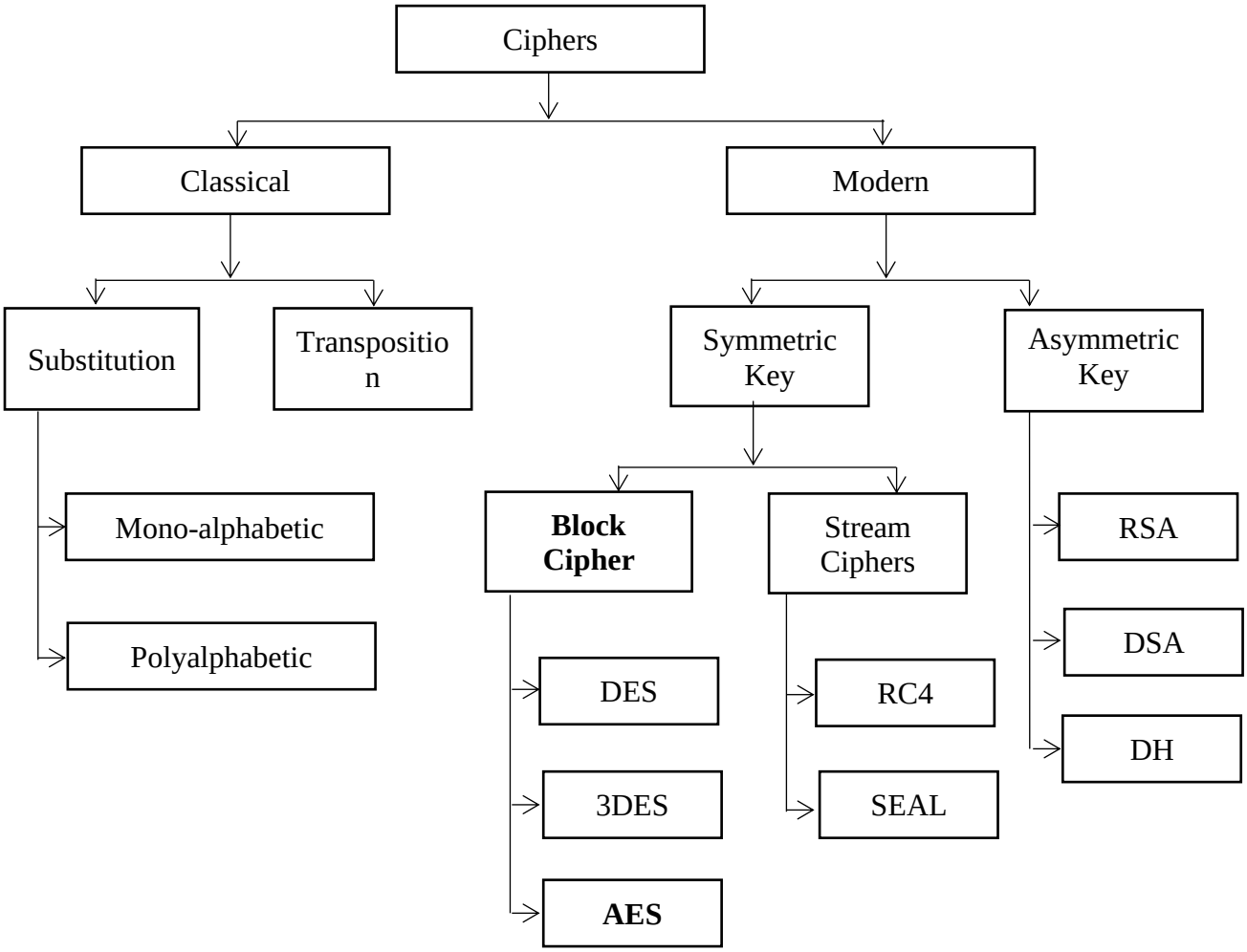

Figure 4: The classification of common encryption algorithms

2. Review of the works of literature

Numerous scholars have looked into the difficulties associated with data security and cryptography. To give a more thorough understanding of the effectiveness of the encryption techniques, we analyze and summarize past work on the subject of data encryption. The criteria considered are processing speed, throughput, power consumption, and avalanche impact.

[13] used a random number generator to enhance the DES algorithm. Here, the message is broken up into blocks of 64 bits, and several keys are created using a 56-bit master key. Using a random number generator, keys are generated. A 56-bit master is used to

Table 2: Various Cryptanalytic Attacks

Type of Cryptanalytic Attack	Information Known to Attacker
Cipher text only	Crypt algorithm the unencrypted text in part
Known-plaintext	Crypt algorithm Unreadable text samples of plaintext and encrypted text, respectively

Chosen-plaintext	Crypt algorithm
	Unreadable text
	Attacker-selected pair of plaintext and its encryption text
Chosen-ciphertext	Crypt algorithm
	Unreadable text
	pair of selected ciphertext and its plaintext encryption
Chosen text	Crypt algorithm
	Unreadable text
	Attacker-selected pair of plaintext and its ciphertext
	pair of selected ciphertext and its plaintext encryption
brute-force attack(passive attack)	Attempt every possible key
dictionary attack	tries using a dictionary of common words
*man in middle attack(*active attack)	tries to exchange the keys between two pairs of the security system's connection channel
timing attack	examines how long it takes to build cryptographic methods

produce keys that are likewise 56 bits. Every block of message bits will use a separate key. Despite the fact that security is improved with this method, the operation will still take a long time to complete.

According to the study done by [14] it was shown that several popular symmetric key encryptions on mobile devices use varying amounts of energy. The battery's remaining power is discovered to be at 45% after only 600 Triple-DES encryptions of a 5 MB file, making further encryptions impossible since the battery is fast depleting.

Random Data Encryption Algorithm was proposed by [15] as an extension to the DES. Pseudo-randomized cipher keys for encryption and methods for delivering cipher keys incorporated in the cipher text are new additions to the DES. The RDEA's overall efficiency has been impeded by the random generator sequence length, memory capacity, vulnerability to linear assaults on the SBox, and significant scheduling issues. It introduces a modified technique that uses a secret key chosen at random from a predetermined list. In other words, depending on the amount of system memory available, the secret key for the RDEA system may be thought of as a long bye of 128 for basic scenarios. Making it compatible with the current 64-bit DES algorithm, which requires more resources and time for Brute-force attacks, and making it simple to change the secret key without the need for any extra hardware or software. There might be up to 256 random secret keys. The disadvantage of this approach is that it relies on the memory or hardware available to create the maximum number of cipher keys, which will make brute force attacks more complicated. Additionally, the parameters that impact RDEA systems include the length of the random generator sequence, and the total amount of system memory

[16] proposed Blowfish, a new block cipher with a secret key. A 64-bit block of data is encrypted using blowfish encryption, which uses keys with lengths ranging from 32 to 448 bits. The actual encryption consists of sixteen rounds and revolves around the use of S-Boxes and intricate key

scheduling. The Blowfish's strength comes in the fact that cryptanalysis methods have no impact on it when it is fully rounded. The technique is not resistant to brute force assaults, however, and anything with less than four rounds is vulnerable to cryptanalysis. A Feistel network iterates a simple encryption function 16 times in the proposed approach. The key may be any size up to 448 bits, and the block size is 64 bits. On massive microprocessors, the actual encryption of data is quite effective. However, there must first be a lengthy start-up process before any encryption can begin.

The notion of utilizing DES three times with three distinct keys is presented by [17]. Triple DES is the name of this improved method. Because three keys are used, security is increased. The problem with this is that it takes too long—48 rounds—to encrypt 64 bits, which slows down the method.

[18] proposed the Block Encryption Standard for the transfer of Data (BEST) to satisfy the availability, confidentiality, and integrity objectives of security. The symmetric key encryption strategy serves as the foundation for this novel technique. This new BEST method is a block cipher that splits data into equal-sized blocks and encrypts each block individually using a unique set of mathematical operations known as Key. This method employs a symmetric key approach, or using the same key at both ends, to encode and decode data. The method that changes the format of the key as it is being sent from one end to the other uses additional security precautions. As a result, the problem with key distribution may be solved quickly. The BEST algorithm's protection against Brute-force assaults is another benefit, since the key is often updated during the encryption process. As a result, it will be exceedingly difficult to extract plaintext from the encrypted text even with knowledge of or access to one key. However, the execution time of the BEST algorithm proposal has to be improved. We are unable to provide a convincing analysis of its applicability to picture and speech/audio data due to its restricted scope. There is no BEST hardware implementation and no use of the compression and encryption methods together.

DES has undergone several improvements and served as the foundation for the following approaches in the area of encryption, according to [19]. The Triple-DES is one of its replacements (3DES or TDEA). 48 rounds of 3DES are used to encrypt the data. This method adds three layers of protection, making the data far more secure and resistant to differential cryptanalysis. The 3DES employs inverse and forward encryptions to function. K1, K2, and K3 are used in 3DES encryption. Data is first decrypted using K3, then K2, and finally K1. The performance of 3DES, however, also requires three times as long to encrypt and decode data since it requires traveling through DES three times.

[20] proposed a combination of the DES and Blowfish ciphers. Additionally, DES has been used in combination with other encryption methods. The DES's key generation was to be strengthened by the planned fusion. It first permutes two keys to encrypt a 64-bit data block, then iterates sixteen rounds of crossover encryption using those same two keys before performing a final inverse permutation. Although it provides additional resistance to Brute Force assaults, it still has the same flaws as normal Blowfish. However, adding two keys to the encryption process made it slower.

[19] proposed a brand-new technique for DES key creation. For key creation, they used the odd-even substitution approach. Every step involves applying the odd-even replacement to the 56-bit key. Plans for the study's future include distinct keys for every 64-bit block. Hopefully, more keys will make the algorithm more resistant to all DES attacks that are now known.

[21] examined the RC5, Blowfish, and DES block cipher algorithms using C# software in Visual Studio 2009. On a 3GHz Pentium®4 with 1GB of RAM running Windows XP Professional Version 2002, Service pack 3, the performance of these three methods was evaluated. A collection of input files was used for a comparative comparison of RC5, Blowfish, and DES, and the encryption and decryption times were assessed. According to the results, RC5 is 2.57 times quicker than DES and 1.54 times faster than Blowfish. Additionally, the results show that the performance of the Blowfish algorithm is inversely related to key size; as key size increases, performance decreases and vice versa. When it comes to resource use, RC5 uses 13.9 MB more RAM than Blowfish and 37.46 MB more memory than DES, although CPU usage is almost the same for all three algorithms. Therefore,

the RC5 block cipher algorithm is more efficient and straightforward than the Blowfish & DES block cipher methods. Where a high encryption rate is needed, RC5 is advantageous.

[22] modified the DES algorithm to enhance the encryption of data sent between any two nodes on the network. It is breakable in its current state. The proposed change aims to extend the break time such that by the time the frame is time-stamped as having been transferred, the information has already arrived at its intended location and the appropriate action may be conducted. This significantly improves the DES algorithm's performance. With respect to the key created above, the RSA-related evidence is pretty evident. The DES algorithm's performance may be further improved in a related future study by incorporating knowledge of a few number theory concepts.

[5], [23] compared the different algorithms, including RC6, DES, IDEA, BLOWFISH, and CAST 128. With the help of the IAIK-JCE package and NetBeans IDE 7.0.1, these algorithms are implemented in Java. Execution times are used to compare algorithms. A test reveals that RC6 has the shortest execution time. Both RC6 and BLOWFISH have comparable throughput. IDEA was outperformed by BLOWFISH. IDEA has a higher throughput than DES for decryption, while DES performs better for encryption. IDEA and CAST 128 both have almost the same throughput.

Three algorithms—DES, 3DES, and RSA—were analyzed by [24]. RSA is an asymmetric key cryptography technique, whereas DES and 3DES use symmetric keys. Data security, data encryption time, and throughput requirements have all been considered while analyzing algorithms. Depending on the inputs, various algorithms are performed in different ways. It was determined that 3DES offers significantly higher levels of confidentiality and scalability than DES and RSA, making it suitable even though DES uses less memory, time, and power to encrypt and decrypt data. However, DES's security can be easily breached using brute force techniques, making it the least secure algorithm.

[12] reviewed the performance and efficiency analyses of several block cipher algorithms used in symmetric key cryptography, including DES, 3DES, CAST-128, BLOWFISH, IDEA, and RC2. The input amount of data (in the form of text, audio, and video), encryption and decryption times, the throughput of encryption and decryption for each block cipher, and power consumption have all been taken into consideration when comparing block cipher algorithms. It was determined that the triple phase properties of 3DES cause it to use more power and have worse throughput than DES.

[25] proposed an algorithm known as the "Enhanced Multi-State DES," a cryptosystem similar to DES. To reduce the likelihood of the full 16-round characteristic being vulnerable to differential cryptanalysis, it extends the DES algorithm so that the iterative number of the f function during each sub-full block's 16 rounds is different. Maximizing the complexity of the S-BOX design to get a good avalanche effect because the S-Box architecture uses a 32-bit output and an 8-bit input. (Added 24 bits) To decrypt plaintext, you need 256 * 232 key combinations. Attacking with brute force is challenging. It will be more difficult for the intruder to detect the actual information by increasing the number of states for presenting the information as well as the number of combinations of the information. The plaintext message, not any specific pattern, determines the value that is extracted from the hash table. A new operation that makes use of two 4 states keys replaces the XOR operation. The four-state key (0,1,2,3,) was used to swap out the two-state key (0,1). This method provides security but increases complexity.

[26] have combined the cryptanalysis of the simplified data encryption standard method with meta-heuristics, and they concluded that the results were poorer than a random search when compared to genetic algorithms.

Cloud Data Sharing Using Cipher Proxy Re-encryption and Ciphertext-Policy Attribute-Based Encryption was proposed by [27]. The proposed system is an attribute-based ciphertext encryption strategy that assigns responsibility for attribute revocation to a cloud server via proxy re-encryption. The access structure for secret sharing schemes (LSSS) is not necessary for the proposed system. Their suggested approach is resistant to assault from cloud servers and unauthorized users. Sharing cloud storage carries the danger of service-related data leakage. Data shared on cloud storage is

encrypted by the data owner to safeguard it, and only authorized users can decode the cloud data. The cloud's privacy and confidentiality limitations make data theft more likely because cloud providers typically have direct access to data and can use it for illicit purposes. Data is kept "in the open," which gives nefarious users a ton of opportunities to steal data.

[28] suggested improving the Data Encryption Standard (DES) key generation algorithm by suggesting key creation using two eight-bit arrays. Additionally, it implies that issues involving weak and semi-weak signals may be fully handled using the approach given. Because of the random number array used in this new key approach, they have increased security without sacrificing efficiency. Better security and performance against attackers is the aim of building an encryption method. The array's size may be increased to make it more challenging to locate. The restriction is that it takes 16 cycles to encrypt and decode data, which slows down the operation. It generates keys using random numbers and sometimes may not be confidential. As it creates 16 random keys, it uses more storage.

[29] proposed visual cryptography depending on some complex algorithms like RSA and the AlGamal which have been offered as a basis for visual cryptography that offers more security than more conventional approaches that rely only on the XOR operation. Producing multiple shares to make it difficult for hackers to obtain the original image, using a proposed permutation table to achieve the diffusion based on the partitioned image into blocks to encrypt the image, and then contacting blocks in decryption algorithm to retrieve the original image, rather than stacking shares to obtain the original image as in the case of traditional methods. This proposal dealt with color and gray images as opposed to the traditional one, which processed only the binary image. The restriction is that if a hacker knows these numbers, they may defeat the method and get the plain picture. Examples of these values are (p, a,) for AlGamal and (p, q, e, d) for RSA.

According to [30] the DES approach maintains safe data transfer while maintaining the message's legitimacy and honesty. In doing so, the message is encrypted before the data broadcasting process begins. Data is encrypted and decrypted using the industry-standard data encryption technique (DES). Since DES is today seen as an unsafe encryption method for various applications, such as financial systems. Various analytical findings show that the cipher's theoretical flaws. Therefore, it becomes crucial to improve this algorithm by giving it additional levels of security. By changing the function implementation, and S-box design, and swapping out the old XOR with a new operation, we may alter this method in the future.

[31] displayed the Twofish algorithm's performance in parallel, which was assessed based on execution time, speedup, and efficiency for varied data sizes and different processor counts. In this study, parallel Twofish was developed in C++ using the open MPI library and run on the IMAN1 supercomputer. According to the results, parallel Twofish has faster execution times for large data sizes compared to small data sizes, but using a large number of processors on a small data size will result in longer running times rather than faster execution times because there will be a significant amount of communication between processors. The findings of the experiment indicate that when there are eight processors, the running time will be reduced and the speed-up of the encryption and decryption operations will be boosted.

According to[32] The Blowfish algorithm is the best method to employ for picture encryption when compared to Twofish, according to the comparison between Blowfish and Twofish, which showed that the latter required more time for image encryption and decryption.

According to the study done by[33], the blowfish algorithm is quicker than RSA when it comes to encrypting or decrypting data. Against encryption and decryption operations, the number of characters will be encrypted and decrypted. According to test findings, blowfish's encryption and decryption processes are both 178,958% and 420.44188% quicker than RSA, respectively. In contrast, RSA performs 63.131% less slowly throughout the encryption procedure than blowfish. Additionally, the decryption procedure performs 80.3399% slower than blowfish.

The practical study carried out by [34] suggests that Blowfish have the greatest speed among the symmetric ciphers. However, if a trade-off in time efficiency is acceptable, it is proved by theoretical research that 3-DES is the least straightforward to break, making it extremely secure for communication. However, given the large number of rounds in 3-DES, it is the slowest cipher.

[35] proposed a new image encryption method that applies the stochastic random blocks and dynamic S-boxes techniques. The image is scrambled and diffused simultaneously in the scheme, which can produce uniformly diffused values and sufficient scrambling for effective pixel shuffling. The simulation demonstrates how effectively the algorithm can cipher plain images into unrecognizably encrypted ones. The security assessment shows that the method offers a satisfying level of security and has the advantages of a large key space, anti-differential attack, and good data loss resistance, as well as high efficiency and good statistical performances when compared to other image encryption methods. The use of a dynamic S-box and spatiotemporal chaos is a flaw in this work. To create a more effective image encryption algorithm, the structure of the dynamic S-box and spatiotemporal chaos must be improved. Additionally, the encryption algorithm needs to be expanded for use in real-world applications, particularly for information cryptosystems that demand both high security and speed.

[36] proposed a novel idea to protect images against attacks. Applying conformal mapping to the private data completes the basic level. The (RSA) technique is then used to encrypt and decode the picture output. The message is concealed within the cover picture using the least significant bit (LSB) concealing technique. Finally, they suggest using GZIP to compress the picture. But using an asymmetric technique to encrypt huge data might slow down the system's processing speed.

[37] proposed a quantum logistic image cryptosystem. RSA and SHA-3 are combined in a quantum logistic image cryptosystem. First, key pairs with private keys and public keys are generated at random using the RSA algorithm. The confusion is then resolved with a fixed matrix. The clear message is then safely stored after the pre-processed image is hashed using the SHA-3 function. The encrypted message can be performed identically to the original message using the RSA algorithm. The initial conditions of the quantum logistic map are computed using a novel mathematical method after mixing both the original and the encrypted messages. The use of an asymmetric algorithm, which is a challenging algorithm to use to encrypt big data, is this paper's main flaw.

The literature review is summarized for comparison based on factors such as technique used, algorithm used, key encryption type research description and their constraints. The summary is shown in Table 3.

Table 3: Review of related works

Author	Technique used	Algorithm used	Key encryption type	Description	Limitation
[13]	Cryptography	DES	Private Key Encryption	used a random number generator to enhance the DES algorithm	The procedure still takes a long time to finish.
[14]	Cryptography	Triple-DES	Private Key Encryption	To show that several popular symmetric key encryptions on mobile devices use varying amounts of energy.	Triple-DES encryptions of a 5 MB file, making further encryptions impossible since the battery is fast depleting.
[15]	Cryptography	Random Data Encryption Algorithm	Private Key Encryption	Random Data Encryption Algorithm was proposed, as an extension to the DES	The disadvantage of this approach is that it relies on the memory or hardware available to create the maximum number of cipher keys,

[16]	Cryptography	Blowfish	Private Key Encryption	proposed Blowfish, a new block cipher with a secret key. A 64-bit block of data is encrypted using blowfish encryption, which uses keys with lengths ranging from 32 to 448 bits.	there must first be a lengthy startup process before any encryption can begin.
[17]	Cryptography	Triple DES	Private Key Encryption	Presented The notion of utilizing DES three times with three distinct keys.	The problem with this is that it takes too long—48 rounds—to encrypt 64 bits, which slows down the method.
[18]	Cryptography	Block Encryption Standard for Transfer of Data (BEST)	Private Key Encryption	This new BEST method is a block cipher that splits data into equal-sized blocks and encrypts each block individually using a unique set of mathematical operations known as Key	the execution time of the BEST algorithm proposal has to be improved
[19]	Cryptography	Triple DES	Private Key Encryption	48 rounds of 3DES are used to encrypt the data. This method adds three layers of protection, making the data far more secure and resistant to differential cryptanalysis.	requires three times as long to encrypt and decode data since it requires traveling through DES three times.
[20]	Cryptography	DES Blowfish	Private Key Encryption	The DES's key generation was to be strengthened by the planned fusion	adding two keys to the encryption process made it slower
[19]	Cryptography	DES	Private Key Encryption	brand-new technique for DES key creation. For key creation, they used the odd-even substitution approach.	same keys for every 64-bit block
[21]	Cryptography	RC5, Blowfish, DES	Private Key Encryption	Evaluated the performance of these three ciphers	performance of the Blowfish algorithm is inversely related to keysize; as keysize increases, performance decreases
[23]	Cryptography	RC6, DES, IDEA, BlowFish, CAST-128	Private Key Encryption	compared the different algorithms and their execution time	RC6 has the shortest execution time
[24]	Cryptography	DES, 3DES, RSA	Private Key Encryption Public Key Encryption	Analyzed the three cryptographic algorithms considering data security, data encryption time, and throughput requirements	DES's security can be easily breached using brute force techniques, making it the least secure algorithm.

[12]	Cryptography	DES, 3DES, CAST-128, BlowFish, IDEA, RC2	Private Key Encryption	reviewed the performance and efficiency analyses of the several block cipher algorithms considering the input amount of data, encryption and decryption times, the throughput of encryption and decryption for each block cipher, and power consumption	the triple phase properties of 3DES cause it to use more power and have worse throughput than DES.
[25]	Cryptography	Enhanced Multi-State DES	Private Key Encryption	proposed an algorithm known as the "Enhanced Multi-State DES", it extends the DES algorithm so that the iterative number of the f function during each sub-full block's 16 rounds is different.	provides security but increases complexity
[26]	Cryptography	simplified data encryption standard and meta-heuristics	Private Key Encryption	combined the cryptanalysis of the simplified data encryption standard method with meta-heuristics	the conclusion was that the results were poorer than a random search when compared to genetic algorithms
[27]	cyber security	Ciphertext-policy attribute-based encryption scheme	access control mechanism	The proposed system is an attribute-based ciphertext encryption strategy that assigns responsibility for attribute revocation to a cloud server via proxy re-encryption.	The cloud's privacy and confidentiality limitations make data theft more likely because cloud providers typically have direct access to data. Data is kept "in the open," which gives nefarious users a ton of opportunities to steal data.
[28]	Cryptography	DES	Private Key Encryption	suggested improving the Data Encryption Standard (DES) key generation algorithm by suggesting key creation using two eight-bit arrays.	The restriction is that it takes 16 cycles to encrypt and decode data, which slows down the operation. As it creates 16 random keys, it uses more storage.
[29]	Cryptography	AlGamal RSA	Public Key Encryption	proposed visual cryptography depending on some complex algorithms like RSA and the AlGamal	The restriction is that if a hacker knows these numbers, they may defeat the method and get the plain picture. Examples of these values are (p, a,) for AlGamal and (p, q, e, d) for RSA.
[30]	Cryptography	DES	Private Key Encryption	discussed the DES technique for securing data transmission while maintaining the	DES is today seen as an unsafe encryption method for several applications, such as financial systems. Some

				authenticity and integrity of the message.	analytical findings support theoretical claims.
[31]	Cryptography	Twofish	Private Key Encryption	displayed the Twofish algorithm's performance in parallel and sequential in IMAN1 supercomputer using Message Passing Interface (MPI), which was assessed based on execution time, speedup, and efficiency for varied data sizes and different processor counts	using a large number of processors on a small data size will result in longer running times rather than faster execution times because there will be a significant amount of communication between processors.
[32]	Cryptography	Blowfish Twofish	Private Key Encryption	Carried out a comparison between Blowfish and Twofish	Twofish required more time for image encryption and decryption.
[33]	Cryptography	Blowfish RSA	Private Key Encryption Public Key Encryption	conducted a comparative analysis of the encryption and decryption processes used in an Android-based email application using Blowfish and RSA.	RSA performs 63.131% less slowly throughout the encryption procedure than blowfish. Additionally, the decryption procedure performs 80.3399% slower than blowfish.
[34]	Cryptography	RSA, DES, Triple DES, and Blowfish,	Public Key Encryption Private Key Encryption	To determine the optimum technique to employ for encrypting text files, this study compares four widely used encryption ciphers, including RSA, DES, Triple DES, and Blowfish, using a web tool that allows for analyzing numerous encryption-related aspects.	given the large number of rounds in 3-DES, it is the slowest cipher.
[35]	Cryptography	dynamic S-boxes and random blocks schemes	hash function	proposed a unique spatiotemporal chaotic system-based dynamic S-box and random block image encryption technique.	the encryption algorithm needs to be expanded for use in real-world applications, particularly for information cryptosystems that demand both high security and speed.
[36]	Cryptography	ElGamal RSA DES 3DES AES	Public Key Encryption Private Key Encryption	offered a broad review of cryptography, information security, and a comparison of symmetric and asymmetric key algorithms while taking into account variables assuring the attainment	Using an asymmetric technique to encrypt huge data slows down the system's processing speed.

				of efficiency, flexibility, and security.	
[37]	Cryptography	RSA SHA-3	Public Key Encryption hash function	based on SHA-3 and RSA, a quantum logistic image encryption technique was presented.	The use of an asymmetric algorithm, which is a challenging algorithm to use to encrypt big data, is this paper's main flaw.

3. Comprehensive Descriptions of Common Encryption Algorithms

A considerable amount of research has been done in the area of cryptography, which covers some of these needs. The effective implementation of cryptographic algorithms in software and/or hardware is the fundamental concept in this discipline. Here are some examples of symmetric key encryption techniques in use:

A. DES (DATA ENCRYPTION STANDARD)

The Data Encryption Standard (DES), which was first used in 1977, is a symmetric block cipher with a block size of 64 bits and a key size of 64 bits that are based on the Feistel structure. Despite being compromised, DES is still being used by many industries, including the American Bankers Association, and in some security standards, such as the IP Security Architecture (IPSec) standard, to provide data security [36].

To encrypt plain text, DES performs 16 rounds of a Feistel-like encryption technique. 16 keys are generated from the initial key using a key schedule for the subsequent rounds of encryption. Figure 2 displays the block diagram for a single DES round [38].

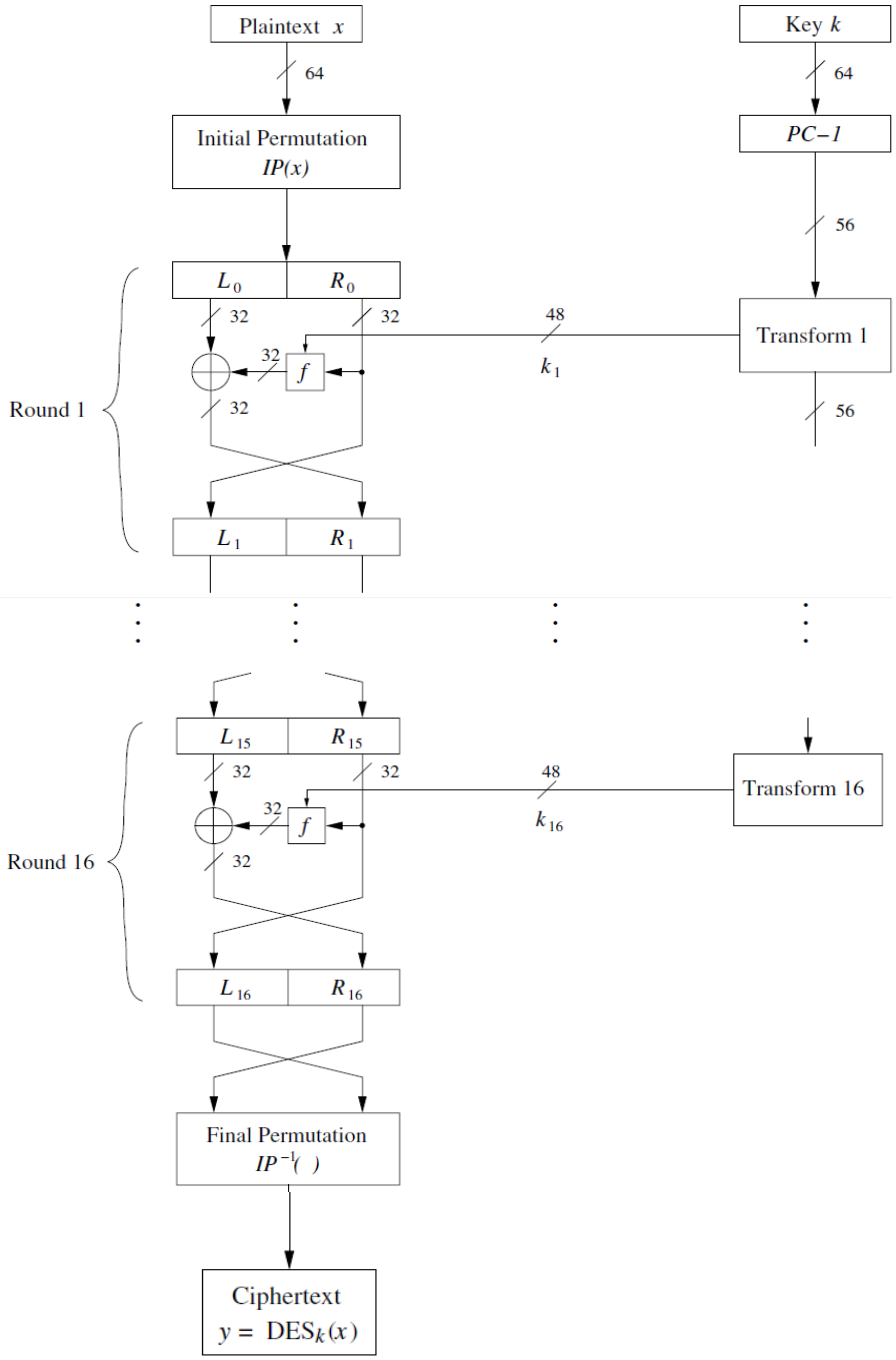

Figure 5: The Feistel structure of DES [39]

DES employs a key that is 64 bits long, but only needs 8 of those bits for odd parity, therefore they are not included in the key length. Since DES uses keys with an effective length of 56 bits, there are 256 potential keys. 256 times as many key combinations are possible with a complete 64-bit key. The 16 encryption rounds of the DES key schedule are not guaranteed to provide random keys in addition to the short key. The produced keys might include all ones, all zeros, or identifiable patterns of ones and zeros [40].

This allowed methods using differential and linear cryptanalysis to attack the DES. Additionally, given the degree of computing power in modern computers, utilizing a brute-force key search does not seem to be too challenging to address the major issues of DES, the Triple DES (3DES) was developed. The plaintext is encrypted using one key in a typical 3DES implementation. After decrypting the cipher text using a different key, it is then encrypted once again using the original key (the first key used). There are two separate keys required to implement the 3DES algorithm. Implementations with three distinct keys are nonetheless also feasible. 3DES provides a key length of 112 bits, which is longer than DES's. This is a 256-combination improvement over the 56-bit key. Although 3DES has a solution for the issue of small keys, it does not completely address the issue of (relatively) non-random key creation. Additionally, 3DES is roughly slow [9]. The disadvantage of DES is that its security may be readily breached and that it only operates quickly on hardware while operating slowly on software [9].

B. 3DES (TRIPLE DATA ENCRYPTION STANDARD)

Without creating a brand-new cryptosystem, Triple DES was created to fix the obvious problems with DES. Triple DES merely increases the DES key size by running the algorithm three times in quick succession with three distinct keys. Since the combined key size is 168 bits (3 times 56), brute-force methods like those employed by the EFF DES Cracker are ineffective. Since the original algorithm was never intended to be used in this way, Triple DES has always been viewed with a certain amount of suspicion. Despite this, no significant design flaws have been found, and it is currently a widely used cryptosystem in many Internet protocols [41].

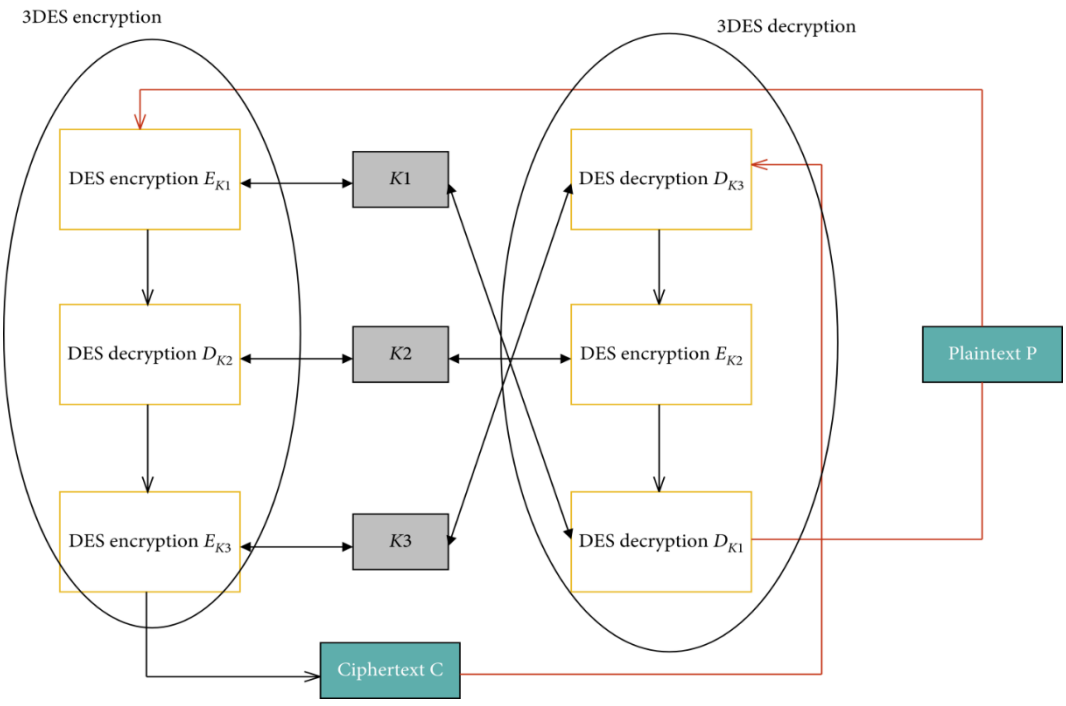

Figure 6: Principle of 3DES encryption algorithm [42]

A cumulative key size of 112 - 168 bits is achieved via Triple-DES, which encrypts data three times as seen in figure 6, while using a new key for each of the three passes. This should result in a strength that is more than adequate to fend off brute force assaults, somewhere around 112 bits. Although triple DES is substantially more powerful than (single) DES, it is quite sluggish in encrypting and decoding blocks of data computationally when compared to other modern block ciphers. As

indicated in Table 4 the National Institute of Standards and Technology (NIST) draft recommendation states that 3DES will be phased out in 2023 and that all users of 3DES should switch to AES as soon as practicable [43].

Table 4: An overview of the standard encryption algorithm.

Cryptographic algorithm	Key length (Bits)	Block Size	Year of Creation	Status
DES	64	64	1973	Obsolete
3DES	64, 128, 192	64	1998	To retire by 2023
AES	128, 192, 256	128	2001	Replacement for DES and 3DES

C. AES (ADVANCED ENCRYPTION STANDARD)

The Rijndael encryption, often known as the Advanced Encryption Standard (AES), was introduced in 2000. In this section, we describe AES and detail its components. DES and 3DES were built on the Feistel network, and higher key and block sizes were among the primary adjustments proposed to implement AES [38]. There are no weak keys in AES, thanks to the key expansion method. A weak key is one that predictably weakens the security of a cipher. For instance, DES is well known to contain insecure keys. Keys that generate identical round keys for each of the 16 rounds are considered weak in DES. When it is made up of alternating ones and zeros, it is an example of a DES weak key. All the round keys in DES become similar as a result of this kind of weak key, which leads to the encryption being self-inverting. In other words, decrypting and decrypting again will result in the same plain content. (DES's few weak keys are quickly identified; therefore it is not thought to be a concern with that cipher.) [38].

Furthermore, a substitution-permutation network in a broader sense is what is used by AES. In AES, word-level permutations are done after each cycle of byte-level replacements [38].

An encryption that focuses on bits is DES. While by utilizing 10, 12, and 14 rounds of a 128, 192, or 256-bit key, byte-oriented encryption AES encrypts data. This offers enhancements of 2^{72}, 2^{136}, and 2^{200} respectively over the 56-bit DES key. Longer keys made it far more difficult to crack the AES.

Table 5: AES algorithm rounds and key size

Number of rounds	Size of the keys
12	128
14	192
16	256

Additionally, the block size of the DES was accounted for by AES. AES uses 128-bit encryption blocks, making it more resistant to information leaks (produced by repeating blocks). With DES, a single key may be used to encrypt up to 32GB of data. On the other side, AES permits the processing of 256 billion gigabytes with the same key before any leaks may happen.

The suggested cipher also incorporates a portion of DES, an iterated block cipher that processes plaintext in rounds, each applying the same overall transformation function to the incoming block; nonetheless, AES is an example of key-alternating block ciphers. In these ciphers, each round begins by applying a diffusion-achieving transformation operation, which might be a mix of linear and

nonlinear steps, to the whole incoming block. This is followed by the application of the round key to the entire block [38]. Figure 3 displays the AES block diagram.

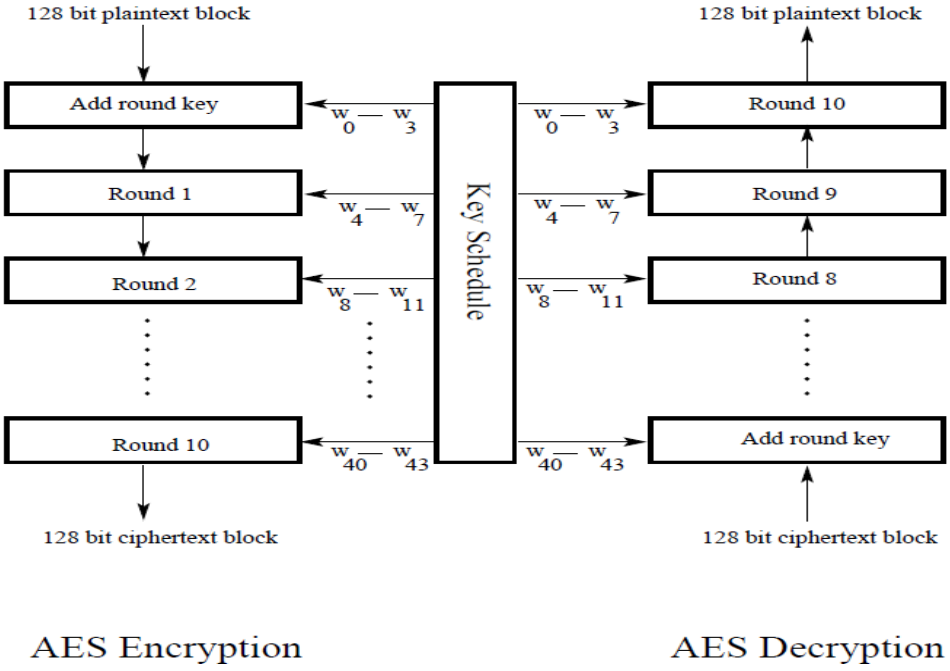

Figure 6: The overall structure of AES for the case of a 128-bit encryption key [38]

Figure 6 depicts the overall architecture of AES encryption and decryption. The input state array is XORed with the first four words of the key schedule before any round-based processing for encryption can start. During decryption, the same thing takes place, but this time the ciphertext state array is XOR with the final four words of the key schedule.
Each cycle of encryption consists of the four stages below: To replace bytes, shift rows, mix columns, and add a round key, are the four methods. The last step is XORing the results of the first three phases with the final four words from the key schedule.
Each decryption cycle includes the following four steps: 1) Invert rows; 2) Invert bytes; 3) Add a round key and 4) Invert columns. The third stage entails XORing four words from the key schedule with the result of the first two processes. You should take note of the variations between the substitution and shifting operations' order in a decryption round and the order in which identical operations are performed in an encryption round.
The "Mix columns" step is skipped in the last round of encryption. The "Inverse mix columns" step is omitted from the final decryption phase.
The many processes that are completed in each round—aside from the last round—are shown in Figure 7.

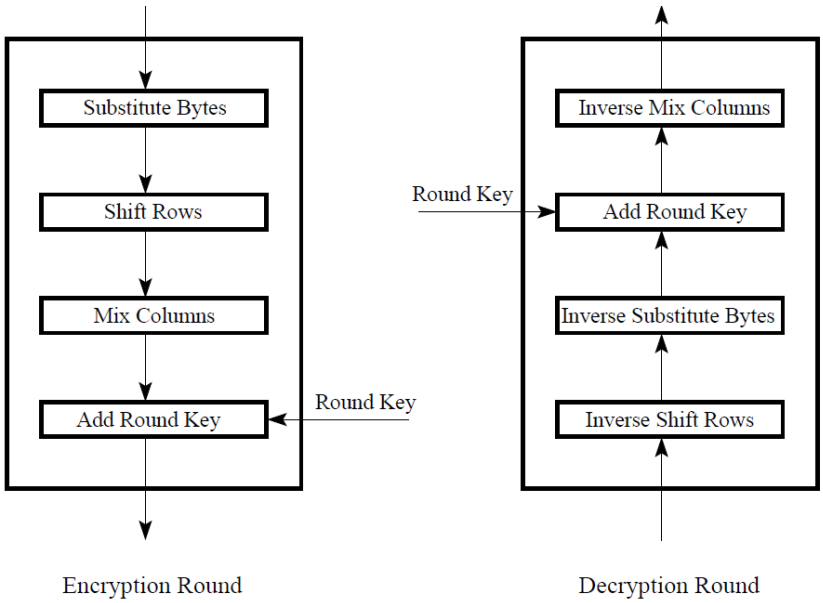

Figure 7: A round of encryption is shown on the left, followed by a round of decryption on the right[38].

STEP 1: (also known as SubBytes since it substitutes bytes one at a time throughout the forward process). (InvSubBytes is the name of the matching replacement step used during decryption.). According to Fig. 8, the AES encryption algorithms pseudocode is shown (Bae et al., 2012). The crucial aspect of the pseudocode is that the "for" loop function does not take into account the last AES round.

```
state = M
AddRoundKey(state, &w[0])
for i = 1 step 1 to 9
        SubBytes(state)
        ShiftRows(state)
        MixColumns(state)
        AddRoundKey(state, &w[i*4])
end for
SubBytes(state)
ShiftRows(state)
AddRoundKey(state, &w[40])
```

Figure 8: Pseudo code for the AES cipher

In this stage, a 16 x 16 lookup table is used to locate a replacement byte in the input state array for a certain byte. The principles of multiplicative inverses in GF(2^8) and bit scrambling are used to eliminate the bit-level correlations inside each byte and produce the entries in the lookup table.

STEP 2: (known as ShiftRows for moving the rows of the state array during the forward phase) (The same transformation during decryption is referred to as InvShiftRows for Inverse Shift-Row Transformation). The purpose of this transformation is to randomly shuffle the order of the bytes inside each 128-bit block.

STEP 3: The third step is known as MixColumns because it involves mixing up the bytes in each column individually during the forward process. InvMixColumns, which stands for the inverse mix column transformation, refers to the corresponding transformation during decryption. To further jumble up the 128-bit input block is the objective here. After 10 rounds of processing, the shift-rows step and the mix-column step make each bit of the ciphertext dependent on each bit of the plaintext. In AES, one bit of the plaintext can have an impact on all 128 bits in the ciphertext block, whereas in DES, one bit of the plaintext typically impacts just 31-bit locations.

STEP 4: (known as AddRoundKey throughout the forward phase to add the round key to the output of the preceding step) (The analogous decryption phase is identified as InvAddRoundKey, which stands for inverse add round key transformation.)

4. AES: Advantages and Efficiency

The key benefits of AES and its improvement over DES are covered in this section. The text block size of AES, which is 128 bits, is its first positive feature (64 bits on DES). The key lengths of 128, 192, and 256 bits are the next. Depending on the key length, the algorithm is referred to as AES-128, AES-192, or AES-256. (Whereas DES uses 56 bits), and round keys are 64 bits. The key elements of AES resemble those of DES but go by different names. For example, RoundKey uses XOR operation; SubBytes employs a sophisticated form of substitution, and ShiftRow functions similarly to permutation by looking up a table and rearranging the bits. The Rijndael Mix Column segment is where the intensive AES computation occurs, and the implementation of Mix Columns is based on a mathematical analysis of the Galois field. The Mix Column transformation affects each column of the 4-by-4-byte matrix created from the input 128-bit data block, just like substitute bytes do. Each of the column's four bytes is translated into a new value that is a function of all four bytes. According to the AES standard scheme, which the US government has adopted as the NIST standard for encryption [43].

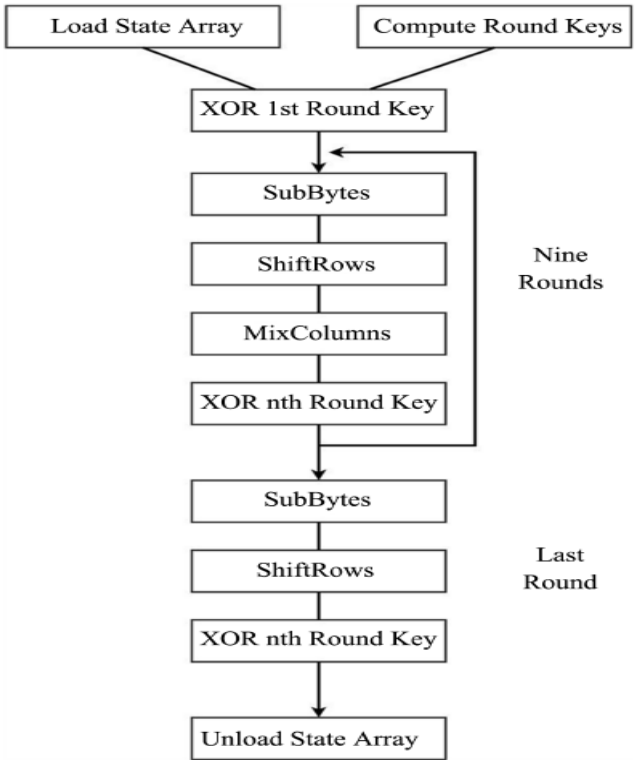

Figure 9: Encryption Structure of each round [43]

Figure 9 depicts what occurs in each round of the AES encryption and decryption process, The Mix Column is not utilized in the final round following AES standards. To add obscurity, the final round only contains the steps Substitute bytes, ShiftRows, and AddRoundKey [43].

The following transformations control how each AES round is conducted.

Substitute Byte transformation

AES uses 128-bit data blocks, which equates to 16 bytes for each data block. The Rijndael Sbox, an 8-bit substitution box, is used in the sub-byte transformation to change each byte (8 bits) in a data block into a different block [38].

Shift Rows transformation

It is a straightforward byte transposition in which, depending on the row location, the bytes in the state's final three rows are cyclically moved. 1-byte circular left shifting is done for the second row. 2-byte and 3-byte left circular shifts are carried out for the third and fourth rows, respectively [38].

Mixcolumns transformation

This round equates to multiplying each column of the states by a matrix. Each column vector receives a fixed matrix multiplier. The bytes in this operation are treated as polynomials rather than integers [38].

Addroundkey transformation

The 128 bits of the current state and the 128 bits of the round key are bitwise XORed. This transformation is an inverse of itself [38].

Even more straightforward than DES and 3DES, the AES algorithm is relatively easy to implement using software and hardware. In comparison to earlier encryption standards like DES and 3DES, AES offered a high level of security against brute force attacks and was faster in terms of computation for a given hardware platform [43]. Bit-level access to the block entering a round is required for the DES substitution step. On the other hand, since AES only performs byte-level operations, software implementation of AES is simple and quick [38].

A. Avalanche impact

A useful characteristic of cryptographic algorithms is the avalanche effect. The output varies by roughly half the bits when an input or key is significantly altered by one bit. Any encryption technique with the ability to cause a big change in the cipher text from a little change in the plaintext or key is said to have an avalanche effect[36].

$$\text{Avalanche Effect} = \frac{Count\ of\ flipped\ bits\ in\ a\ text\ cipher}{\text{Number of bits in the ciphered text}} \quad [44]$$

According to Sanap & More's performance study of encryption algorithms, which used the rigorous avalanche criteria and the avalanche effect as performance indices, the avalanche effect played a crucial role. This crucial performance characteristic for both AES and DES is examined using Cryptool. Only one bit of the input message is altered in this test; the key remains unchanged. 47.24 is the observed parameter for the AES test. AES's avalanche is 48.55 when the key is altered by one bit and the input plaintext is maintained constant. When the key in DES is constant or changed, the avalanche impact is 40.34 and 40.34, respectively. Both AES and DES demonstrate a high avalanche effect, according to the test findings. However, AES has a greater avalanche effect. Additionally, it is established that AES has a greater avalanche effect than DES[44].

B. Speed

K.B. Logunleko, O.D. Adeniji, (2020) experimented to compare the performance of the AES and DES algorithms utilizing PhoneGap, Javascript, HTML5, and mobile devices with a minimum 2.0 GHz CPU, 2 GB of RAM, and 16 GB of storage space. Encryption and decryption time were used as parameters in the performance assessment of the two techniques stated. The study and assessment of the outcome display the average encryption and decryption times for the AES method and DES for data sizes of 15 Bytes. DES encryption and decryption took 0.009 seconds and 0.008 seconds, compared to 0.008 seconds and 0.004 seconds for AES encryption. In a comparable experiment, a 24-byte SMS was used. AES took 0.009 seconds and 0.005 seconds to encrypt and decode data, whereas DES took 0.011 seconds and 0.006 seconds. Another experiment used 41-byte SMS messages, and the encryption and decryption times for AES and DES, respectively, were 0.011 and 0.005 seconds and 0.012 and 0.015 seconds. Similar to this, the data collected during the experiment for 67 bytes of SMS plaintext reveals that AES encryption took 0.016 seconds and decryption took 0.01 seconds, whereas DES took 0.009 seconds and 0.005 seconds, respectively. The experiment's

final results showed that DES encryption and decryption for 103 bytes of SMS plaintext took 0.018 seconds and 0.012 seconds, respectively, whereas AES encryption and decryption took 0.012 seconds and 0.008 seconds.

C. Brute Force Attack

The brute force approach according to [38], involves utilizing a quick guessing tool to test every conceivable combination to locate the encryption key. Thousand trillion keys per second (10^{12} keys/sec) was a pace we took into consideration. We can generate 256 different keys for DES, which uses keys that are 56 bits long. When we divide 2^{56} by 10^{12} and attempt the brute force attack, we can calculate how long it will take to get the correct key. 72058 seconds pass in this instance. A 20-hour result, which is less than a day, is obtained by dividing this value by 3600. If we apply the same procedure to the other encryption techniques, dividing the result by 24 to get the number of days and by 365.25 to obtain the number of years, we arrived at the following conclusion: 1.6×10^{14} years for 3DES (2112 keys). The security of AES is thus unaffected by any known threats since, given the length of time since the cipher's introduction in 2001, its temporal complexity is well beyond what any computer system will be able to manage for a very long time. The worst-case time complexity for a brute-force assault on a key with a 128-bit value, AES, is 10^{19} years (2^{128} keys). A theoretical assault because it cannot be implemented in practice, such a brute-force attack would be regarded as such.

D. Cryptanalysis

It's challenging to do cryptanalysis for a number of reasons. First off, an exhaustive key search is quite difficult due to the huge key size of 128 bits. There was a lot of interest in examining the block ciphers of the day—DES being the most prominent—from the perspective of their vulnerabilities to differential and linear cryptanalysis in the 1990s, which is the decade before the development of the Rijndael cipher, which is the forerunner to the AES standard. AES has a phase called multiplicative inverse (MI) byte substitution that is designed to guard it against this kind of cryptanalysis. It was discovered that block ciphers using SBoxes based on polynomial arithmetic in Galois fields were susceptible to a brand-new attack known as the interpolation attack. The SBox in AES has a bit scrambling component as a defense against the interpolation attack[38]

5. The Extensive Comparative Analysis

A theoretical study of the chosen algorithms was conducted based on reviews of the literature by different scholars. Whereas memory usages, output bytes, and battery consumption are the main issues of concern, encryption techniques play a crucial part in communication security. Performance assessment is carried out using the chosen algorithms DES, 3DES, and AES.

Table 6: Symmetric Encryption Algorithm Comparative Analysis

Algorithms/ Parameters	DES	3DES	AES	References
Key length (bits)	56 bits out of 64 bit	112 bits or 168 bits	128, 192, or 256 bits	[46]; [36]
Cipher Type	Symmetric	Symmetric	Symmetric	[46]; [10]
Block (bits)	64 bit	64 bit	128, 192, or 256 bits	[47];[46]
Rounds	16	16	10, 12 or 14 depending on the key size	[36]
Structure of algorithm	Feistel	Feistel	Non Feistel Substitution-Permutation	[36]; [48]
Developed	1975	1978	1998	[12]; [47]
Security	Inadequate	Passing	High	[46]; [12]

Memory consumption	Higher than AES	Higher than DES	Low	[36]; [46]
Possible Key	2^{56}	2^{112}	2^{128}	[46]; [36]; [12]
Time for the Brute Force key attack	Less than one day	1.6×10^{14} years	10^{19} years	Look up section 4 part C
Avalanche effect	Less compared to that AES	Less compared to that AES	Strong compared to that DES	[36]
Encryption Speed	Higher than AES	Higher than AES	Six times faster than DES	[49]
Time evaluation for 915 Kb of a text file	2133ms	2235ms	2050ms	[47]

6. Conclusion

This paper has presented a detailed and comprehensive comparison of the DES, 3DES, and AES encryption algorithms. Through a thorough examination of experimental studies and theoretical analyses, the strengths and weaknesses of these algorithms have been assessed. The findings of this research provide valuable insights into the performance and security aspects of each algorithm, enabling informed decision making in real-world applications. This study serves as a valuable resource for researchers, practitioners, and policymakers seeking to understand the nuances and trade-offs involved in selecting the most suitable encryption algorithm for their specific needs. AES method was shown to take the least amount of time to encrypt data when compared to the other stated algorithms, based on the text files utilized and the results from reviews made. The implementation of the AES cipher is easier, and security is reinforced with greater key and data block sizes, which are two major areas where it outperforms DES encryption. In addition, we evaluated the computational benefits of the AES data encryption method that may have supported its adoption as the US and now the world standard for block encryption. While AES is substantially more secure than DES and 3DES, it is also computationally faster than DES and 3DES for both hardware and software encryption procedures. Based on a lower number of repeated operations than the previously used standards, DES and 3DES, AES is the best block encryption algorithm. For a particular hardware platform and plaintext, it was discovered that AES is around 1.6 times quicker than DES and 4.8 times faster than 3DES [43]. This study sheds light on how AES became the de facto encryption standard throughout the globe because of its higher security performance against attacks and quicker hardware encryption speed, mostly because of fewer permutation operations. AES, when compared to DES and 3DES has the best security, with a low memory consumption rate. It has a very strong avalanche effect with an encryption speed that is six times faster than DES. AES time evaluation for 915kb of a text file is 2050ms, which is 83ms faster than DES and 185ms faster than 3DES. Therefore, e-initiatives (such as e-voting, e-Commerce, e-banking, etc.) can use the Avanced Encryption Standard for data security to enforce the integrity of data transmitted over an unsecure network.

7. Contributions to Knowledge

The research work carried out a comparative analysis of the symmetric cryptographic algorithm (DES, 3DES, and AES) as a data security tool, and also observed that the AES cryptographic algorithm is better off compared to DES and 3DES.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

Declaration of interest

The authors affirm that they have no known financial or interpersonal conflicts that would have seemed to have an impact on the research presented in this study.

Acknowledgement
The authors of this review article wish to acknowledge the ongoing support provided by Kampala International University, Uganda, towards the ongoing studies of Ubochi Chibueze Nwamouh.

Reference

[1] A. A. Ojugo, "Cryptography: Salvaging Exploitations against Data Integrity," *Am. J. Networks Commun.*, vol. 2, no. 2, p. 47, 2013, doi: 10.11648/j.ajnc.20130202.14.
[2] M. Fadlan, Haryansyah, and Rosmini, "Three Layer Encryption Protocol: An Approach of Super Encryption Algorithm," *3rd Int. Conf. Cybern. Intell. Syst. ICORIS 2021*, 2021, doi: 10.1109/ICORIS52787.2021.9649574.
[3] S. Srinivasan and A. Rothacker, "SpiderLabs Blog Decade Retrospective : The State of," 2022.
[4] L. Petkova, "KNOWLEDGE – International Journal," vol. 49, pp. 469–474, 2021.
[5] P. Patil and R. Bansode, "Performance Evaluation of Hybrid Cryptography Algorithm for Secure Sharing of Text & Images," *Int. Res. J. Eng. Technol.*, vol. 7, no. 9, pp. 3774–3778, 2020.
[6] N. Ahmad and M. Habib, "Analysis of Network Security Threats and Vulnerabilities: by Development & Implementation of a Security Network Monitoring Solution," *Researchgate*, no. January 2010, p. 93, 2010, [Online]. Available: https://www.researchgate.net/publication/202784990_Analysis_of_Network_Security_Threats_and_Vulnerabilities_by_Development_Implementation_of_a_Security_Network_Monitoring_Solution
[7] S. Suwarjono, L. Sumaryanti, and L. Lamalewa, "Cryptography Implementation for electronic voting security," *E3S Web Conf.*, vol. 328, p. 03005, 2021, doi: 10.1051/e3sconf/202132803005.
[8] F. Thabit, S. Alhomdy, and S. Jagtap, "A new data security algorithm for the cloud computing based on genetics techniques and logical-mathematical functions," *International Journal of Intelligent Networks*, vol. 2. pp. 18–33, 2021. doi: 10.1016/j.ijin.2021.03.001.
[9] Baha Eldin Hamouda Hassan Hamouda, "Comparative study of different cryptographic algorithms," *J. Inf. Secur.*, vol. 8, no. 4, pp. 26433–26438, 2020, doi: 10.4236/jis.2020.113009.
[10] M. N. Alenezi, H. Alabdulrazzaq, and N. Q. Mohammad, "Symmetric encryption algorithms: Review and evaluation study," *Int. J. Commun. Networks Inf. Secur.*, vol. 12, no. 2, pp. 256–272, 2020.
[11] C. Tezcan, "Key lengths revisited: GPU-based brute force cryptanalysis of DES, 3DES, and PRESENT," *J. Syst. Archit.*, vol. 124, no. December 2021, p. 102402, 2022, doi: 10.1016/j.sysarc.2022.102402.
[12] C. Riman and P. E. Abi-Char, "Comparative Analysis of Block Cipher-Based Encryption Algorithms: A Survey," *Comput. Fraud*, vol. 3, no. 1, pp. 1–7, 2015, doi: 10.12691/iscf-3-1-1.
[13] B. Schneier, "Applied Cryptography," *Electr. Eng.*, vol. 1, no. [32, pp. 429–455, 1996, doi: 10.1.1.99.2838.
[14] N. Ruangchaijatupon and P. Krishnamurthy, "Encryption and Power Consumption in Wireless LANs," *Third IEEE Work. Wirel. LANS*, pp. 148–152, 2001.
[15] M. Ammar, A., El Sherbini, A., Ashour, I., & Shiple, "FUndtim Data," *Ammar, A., El Sherbini, A., Ashour, I., Shiple, M. (2005). Random data encryption algorithm (RDEA). Proc. Twenty-Second Natl. Radio Sci. Conf. 2005. NRSC 2005.*, no. Nrsc, pp. 359-366., 2005, doi: DOI:10.1109/NRSC.2005.194020.
[16] B. Sctmeier, "Description of a New Variable-Length Key, 64-Bit Block Cipher (Blowfish)," *Anderson, R. Fast Softw. Encryption. FSE 1993. Lect. Notes Comput. Sci. vol 809. Springer, Berlin, Heidelb.*, vol. 809, pp. 157–158, 2005, doi: 10.1007/3-540-58108-1_24.
[17] W. C. Barker, "NIST SP 800-67: Recommendation for the Triple Data Encryption Algorithm," vol. 1, pp. 1–40, 2008.
[18] A. Kaushik, M. Barnela, and A. Kumar, "Block encryption standard for transfer of data," *ICNIT 2010 - 2010 Int. Conf. Netw. Inf. Technol.*, pp. 381–385, 2010, doi: 10.1109/ICNIT.2010.5508489.
[19] A. M. Sison, B. T. Tanguilig, B. D. Gerardo, and Y. C. Byun, "Implementation of improved des algorithm in securing smart card data," *Commun. Comput. Inf. Sci.*, vol. 340 CCIS, no. c, pp. 252–263, 2012, doi: 10.1007/978-3-642-35267-6_33.
[20] A. H. Al-Hamami, M. Alaa, and H. Al-Hamami, "A proposed Modified Data Encryption Standard algorithm by Using Fusing Data Technique," *World Comput. Sci. Inf. Technol. J.*, vol. 1, no. 3, pp. 88–91, 2011, [Online]. Available: https://www.researchgate.net/publication/283722089
[21] H. K. Verma and R. K. Singh, "Performance Analysis of RC5, Blowfish and DES Block Cipher Algorithms," *Int. J. Comput. Appl.*, vol. 42, no. 16, pp. 975–8887, 2012, [Online]. Available: https://pdfs.semanticscholar.org/cdb7/b397e365338f7ddf78110c7ef7e190afca0e.pdf
[22] A. K. Santra and S. Nagarajan, "A Modified DES and Triple DES Algorithm for Wireless Networks," *Int. J. Comput. Sci. Netw. Secur.*, vol. 13, no. 4, p. 44, 2013.
[23] K. Aggarwal, J. Kaur Saini, and H. K. Verma, "Performance Evaluation of RC6, Blowfish, DES, IDEA, CAST-128 Block Ciphers," *Int. J. Comput. Appl.*, vol. 68, no. 25, pp. 10–16, 2013, doi: 10.5120/11749-7244.
[24] S. T. Marwaha Mohit, Bedi Rajeev, Singh Amritpal, "COMPARATIVE ANALYSIS OF CRYPTOGRAPHIC ALGORITHMS," *Int. J. Adv. Eng. Technol.*, pp. 16–18, 2013, doi: 10.22214/ijraset.2022.43370.
[25] P. Patel, K. Shah, and K. Shah, "Enhancement of DES Algorithm with Multi State Logic," *Int. J. Res. Comput. Sci.*, vol. 4, no. 3, pp. 13–17, 2014, doi: 10.7815/ijorcs.43.2014.085.
[26] F. Teytaud and C. Fonlupt, "A Critical Reassessment of Evolutionary Algorithms on the Cryptanalysis of the Simplified

Data Encryption Standard Algorithm," *Int. J. Cryptogr. Inf. Secur.*, vol. 4, no. 2, pp. 1–11, 2014, doi: 10.5121/ijcis.2014.4201.

[27] D. R. S. StanleyRaja S. J., "Cloud Data Sharing Using Cipher Proxy Re-encryption and Ciphertext-Policy Attribute Based Encryption," vol. 3, no. 3, pp. 222–229, 2016.

[28] D. Rani Bansal and P. Thakur, "Improved Key Generation Algorithm In Data Encryption Standard (DES)," vol. 3, no. 2, 2016, [Online]. Available: www.ijiras.com

[29] A. Kadhim and R. M. Mohamed, "Visual cryptography for image depend on RSA & AlGamal algorithms," *Al-Sadiq Int. Conf. Multidiscip. IT Commun. Tech. Sci. Appl. AIC-MITCSA 2016*, pp. 195–200, 2016, doi: 10.1109/AIC-MITCSA.2016.7759935.

[30] N. Kaur and S. Sodhi, "Data Encryption Standard Algorithm (DES) for Secure Data Transmission," *Int. J. Comput. Appl.*, no. Icaet, pp. 975–8887, 2016.

[31] H. Harahsheh and M. Qatawneh, "Performance Evaluation of Twofish Algorithm on IMAN1 Supercomputer," *Int. J. Comput. Appl.*, vol. 179, no. 50, pp. 1–7, 2018, doi: 10.5120/ijca2018916654.

[32] C. L. Chowdary, P. Nallamothu, M. C. Reddy, and B. Vijay, "Comparative Study on Blowfish and Twofish Algorithms for Image Encryption and Decryption," pp. 941–945, 2020.

[33] D. Y. SYLFANIA, F. P. JUNIAWAN, LAURENTINUS, and H. A. PRADANA, "Blowfish–RSA Comparison Analysis of the Encrypt Decrypt Process in Android-Based Email Application," vol. 172, no. Siconian 2019, pp. 113–119, 2020, doi: 10.2991/aisr.k.200424.017.

[34] S. S. and C. M. Nidhi Girish, Pranav B, "Journal of Computer Engineering & Information Technology Comparative Analysis of Encryption Algorithms Against Text Files," vol. 9, no. 3, 2020, doi: 10.37532/jceit.2020.9(3).226.

[35] X. Wang and J. Yang, "A novel image encryption scheme of dynamic S-boxes and random blocks based on spatiotemporal chaotic system," *Optik (Stuttg).*, vol. 217, p. 164884, 2020, doi: 10.1016/j.ijleo.2020.164884.

[36] L. A. Zahraa Ch. Oleiwi, Wasan Alawsi, Wisam Alisawi, Ali Alfoudi, "Overview and Performance Analysis of Encryption Algorithms," *J. Phys. Conf. Ser.*, vol. 1664, no. 1, 2020, doi: 10.1088/1742-6596/1664/1/012051.

[37] G. Ye, K. Jiao, and X. Huang, "Quantum logistic image encryption algorithm based on SHA-3 and RSA," *Nonlinear Dyn.*, vol. 104, no. 3, pp. 2807–2827, 2021, doi: 10.1007/s11071-021-06422-2.

[38] Avi Kak, "Lecture 8 : AES : The Advanced Encryption Standard Lecture Notes on ' Computer and Network Security," no. 3, 2022, [Online]. Available: https://engineering.purdue.edu/kak/compsec/NewLectures/Lecture8.pdf

[39] Paar Christof;Jan Pelzl, *Understanding Cryptography.A textbook for Students and Practictioners*, vol. 53, no. 9. 2010. [Online]. Available: file:///C:/Users/User/Downloads/fvm939e.pdf

[40] B. Kapoor and P. Pandya, "Data Encryption," *Comput. Inf. Secur. Handb.*, pp. e83–e107, 2017, doi: 10.1016/B978-0-12-803843-7.00046-6.

[41] J. Heyszl *et al.*, "Investigating profiled side-channel attacks against the DES key schedule," *IACR Trans. Cryptogr. Hardw. Embed. Syst.*, vol. 2020, no. 3, pp. 22–72, 2020, doi: 10.13154/tches.v2020.i3.22-72.

[42] M. Chen, "Accounting Data Encryption Processing Based on Data Encryption Standard Algorithm," *Hindawi Complex.*, vol. 2021, 2021, doi: 10.1155/2021/7212688.

[43] C.-W. Cheng, M. H. Cantu, and S. Kumar, "Analyzing Computational Components of Standard Block Encryption Schemes," *J. Comput. Commun.*, vol. 10, no. 06, pp. 81–89, 2022, doi: 10.4236/jcc.2022.106007.

[44] S. D. Sanap and V. More, "Performance analysis of encryption techniques based on avalanche effect and strict avalanche criterion," *2021 3rd Int. Conf. Signal Process. Commun. ICPSC 2021*, no. May, pp. 676–679, 2021, doi: 10.1109/ICSPC51351.2021.9451784.

[45] A. M. L. K.B. Logunleko, O.D. Adeniji, "A Comparative Study of Symmetric Cryptography Mechanism on DES, AES and EB64 for Information Security Article," *Int. J. Sci. Res. Comput. Sci. Eng.*, vol. 8, no. 1, pp. 45–41, 2020, [Online]. Available: https://www.researchgate.net/publication/342242454

[46] Binita Thakkar, "A Survey for Comparative Analysis of various Cryptographic Algorithms used to Secure Data on Cloud," *Int. J. Eng. Res.*, vol. V9, no. 08, pp. 753–756, 2020, doi: 10.17577/ijertv9is080328.

[47] I. H. Latif, "Time Evaluation of Different Cryptography Algorithms Using Labview," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 745, no. 1, 2020, doi: 10.1088/1757-899X/745/1/012039.

[48] P. Kumar and P. Singhal, "A Comparative Study of AES vs DES-A Review," vol. XI, no. 420, pp. 420–424, 2019.

[49] A. Sukiatmodjo and Y. D. Setianto, "Speed and Power Consumption Comparison between DES and AES Algorithm in Arduino," *Sci. J. Informatics*, vol. 6, no. 1, pp. 45–53, 2019, doi: 10.15294/sji.v6i1.17838.