Alfarama Journal of Basic & Applied Sciences

Faculty of Science Port Said University

https://ajbas.journals.ekb.eg
ajbas@sci.psu.edu.eg
http://sci.psu.edu.eg/en/

July 2023, Volume 4 , Issue III

DOI:https://doi.org/10.21608/ajbas.2022.141544.1105

ISSN 2682-275X

Submitted: 30/05/2022
Accepted: 12/12/2022

Pages: 433 - 455

Review Article: Cryptographic Algorithms for Enhancing Security in Cloud Computing

Doaa.S.El-Morshedy [1,*], Noha.E.El-Attar [2], Ibrahim.M.Hanafy [1], Wael.A.Awad[3]

[1] Faculty of Science, Port Said University, Port Said 41523, Egypt
[2] Faculty of Computers and Artificial Intelligence, Benha University, Benha 13518, Egypt
[3] Faculty of Computers and Artificial Intelligence, Damietta University, Damietta 34711, Egypt

***Corresponding author: doaa_morshady@yahoo.com**

ABSTRACT

Cloud computing allows a huge amount of data storage and processing power to be available to users over the Internet. Many organizations are migrating from traditional data storage to cloud storage, which provides an efficient method to access data from anywhere and at any time. However, organizations' biggest barrier to adopting cloud computing is data security. Data security is one of the most critical aspects of cloud computing. As a result, there are various data security methods and implementations. Data encryption is the most commonly used method for protecting data security, which means that encrypting data before uploading it to the cloud prevents unauthorized people from accessing it. This article provides an overview of existing symmetric and asymmetric cryptography algorithms. We cover Advanced Encryption Standard Algorithm (AES), Data Encryption Standard (DES), Triple Data Encryption (TDES), Twofish, and Blowfish for symmetric encryption techniques. Rivest - Shamir Adleman (RSA), Diffie-Hellman Key Exchange (DHKE), and El-Gamal are handled as asymmetric encryption algorithms.

Key Words:

Asymmetric key cryptography; Cryptography; Symmetric key cryptography.

I. INTRODUCTION

Cloud computing is a term that may be defined as a platform to provide several computing services; the most well-known is storage. Individuals and organizations can download data from online inventories and repositories directly into their devices. Similarly, they can equip all of their devices with the specifications they need by synchronizing them. Cloud computing's service, which provides individuals and institutions with fast transaction and data transmission speeds, is packaged into a system that incorporates network parts, multiple servers, massive storage spaces, and many applications [1]. Cloud computing validated the identity and authenticity of the person requesting access to a cloud service. So to access the data stored on the cloud server, the user must have privileges and be registered [2]. The data must be in a set format specified by the provider; as a result, the provider knows where the users' data is placed and has complete access privileges to the data. Protecting data is a complicated task in the Cloud. The brokers who have many permissions to access the users' data may affect the confidentiality of data

stored in the Cloud. Thus, maintaining data confidentiality is even more important in a cloud environment [3]. Many cloud providers protect the user's data confidentiality by encoding all data before storing it on cloud Servers [4]. So, in data security, cryptography plays a significant role [5].

Cryptography is essentially the process of hiding information so that only authorized people can access it [6]. Converting plain text data into an incomprehensible ciphertext by the encryption process and then reversing it to the original data using a decryption process based on several algorithms [7]. As a result, using an appropriate encryption and decryption mechanism may assure data security and privacy in a cloud computing environment [8]. Cryptography is classified into two major categories: symmetric and asymmetric [9]. Both asymmetric and symmetric encryption technologies provide a mechanism for hiding communication contents from prying eyes. their primary functions are widely differing [10].

In symmetric cryptography, only one key, called the public key, is utilized to encrypt and decrypt data between both the sender and the receiver. An asymmetric key completes the encryption and decryption operations using two keys: public keys for encryption and private keys for decryption. Symmetric algorithms are over one thousand times quicker than asymmetric algorithms because they need fewer process resources [11].

While most cryptosystems operate primarily between two parties (sender and receiver), some modifications extend their capabilities beyond the standard. These algorithms have variants that allow encryptions to be processed between multiple parties[12].

In general, cryptographic techniques have been used for many aspects, such as:

a) Access Control, which refers to the unique verification of the group authorized to connect to the delivered message using proper authentication.

b) Data Integrity controls database access for a certain group or individual.

c) Non-repudiation, where the sender and the recipient agree that the report will be acknowledged.

d) Authentication, is the process of allowing a certain individual's identification.

e) Confidentiality, the basic purpose of the encryption and decryption process is to ensure that the cipher key is only owned by the message's receiver [6].

This paper addresses the most popular cryptographic algorithms utilized in the Cloud computing environment and recognizes their benefits and shortages in a literature review. The remainder of this paper is structured as follows: Section 2, addresses some of the common cryptographic classification techniques. In Section 3, a comparison analysis between some cryptographic algorithms (i.e., AES, DES, TDES, Blowfish, Twofish, RC2, El-Gamal, DHKE, and RSA) is presented. Section 4, is an analysis of the related studies that address the security aspects of cloud computing are discussed. Finally, Section 5, concludes this study by suggesting some future work.

2. CRYPTOGRAPHIC TECHNIQUES CLASSIFICATION

There are three types of cryptographic algorithms: hash algorithms, symmetric algorithms, and asymmetric algorithms, as indicated in Figure1.

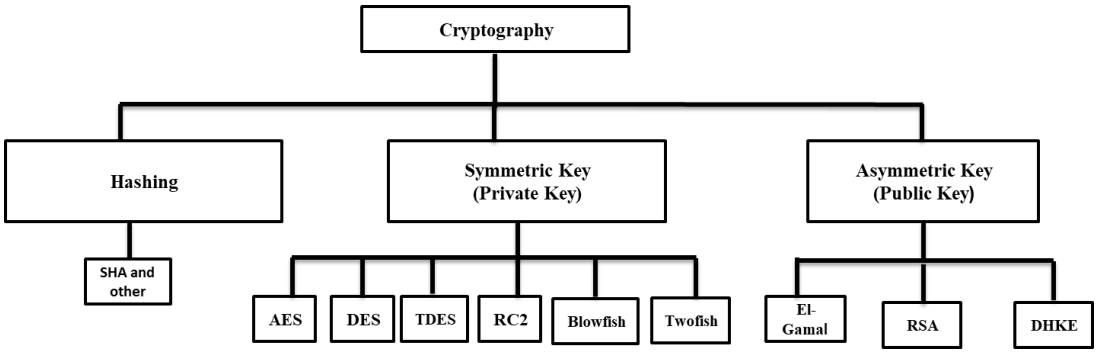

Figure.1 Classification of cryptography techniques [11]

2.1 Hash Functions

The goal of cryptographic hashing is to verify the data and the sender's integrity and authenticity. This method produces a value (as a digest) for the data using a hash function. At the recipient's end, this hash value is examined to determine whether anything has changed. RIPEMD (RACE Integrity Primitives Evaluation Message Digest) and SHA are the most widely used hash algorithms (Secure Hash Algorithm). [13].

2.2 Symmetric Encryption Algorithms

In the encryption/decryption operations, symmetric algorithms employ a single shared secret key. [14], as shown in Figure 2.

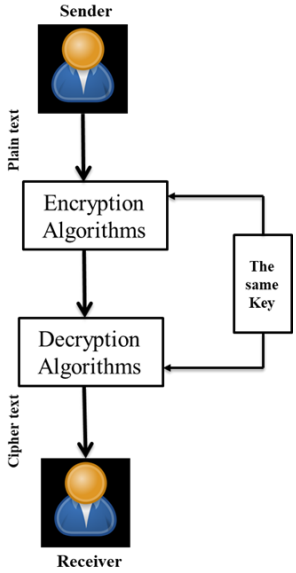

Figure 2. The general idea of symmetric-key cryptography [6]

Traditionally, symmetric encryption systems have been divided into stream and block ciphers. A block cipher divides the plaintext into fixed-size blocks, such as 64 or 128 bits, then encrypts each individually using the same key-dependent transformation. It is also a function that may be reversed [20]. There are two kinds of stream ciphers: For synchronous stream ciphers (i.e., the key stream is based on the key) and asynchronous stream ciphers (i.e., the ciphertext is dependent on the key). Each bit in a stream cipher is encrypted independently [15].

Blowfish, Serpent, IDEA, Twofish, Threefish, DES, TDES, and AES are examples of block cipher algorithms. In this type of algorithm, the size of the key used in an encryption scheme determines its strength. When the key size is large, the encryption difficulty rises. Foa given algorithm, the encryption process with a bigger key is more difficult to crack than encryption with a smaller key. The length of the key varies based on the algorithm. For instance, the DES algorithm uses a 64-bit key, whereas AES uses a 128,192and 256-bit key, RC2 utilizes a 64-bit key, and blowfish utilizes a 32-448-bit key [16].

2.2.1 Advanced Encryption Standard Algorithm (AES)

AES is a symmetric encryption method created by Joan Daemen and Vincent Rijmen in 2001. The AES algorithm is successful in achieving data confusion and dissemination. Because this method is performed on binary data, it can be processed quickly. It's usually used in protecting government data, network security, and computer security [18]. Nowadays, AES is widely used to encrypt data in a secure way worldwide [19].

The key lengths of AES are 128, 192, and 256 bits respectively. For 128-bit keys, for10 rounds are required, for 192-bit keys, 12 rounds, and for 256-bit keys, 14 rounds are required [20]. The four steps of

the AES algorithm's encryption processes are Byte-Substitution, Shift-Rows, Mix-Columns, and Add Round-Key.

a) Key expansion: For each group of rounds, AES uses a useful key scheduling mechanism that creates a 128-bit unique key. This technique performs actions on each word that is 32 bits long. As a consequence, each key includes the following four words:

1) The final word of the last round is rounded eight bits to the left.
2) The S-box replaces the four bytes resulting from the first word in the last round;
3) The four bytes resulting XORed the first word of a previous round.
4) The first byte of each of the four-bit results is XORed with the round constant (Rcon), which differs with each round [21].

Except for the last round of AES, each round of AES consists of the same operations on the state matrix to be executed. The operations are as follows:

b) Add round-Key: It is the first step in the encryption operation, and it is required in every round. The final round includes all of the operations from the previous rounds except Mix Columns [22], [23]. To generate a 128-bitstate array output, a 128-bit plaintext array is XORed with a 128-bit key [21].

c) Bytes to Substitute: These procedures are based on a substitution box designed specifically for this purpose. The fundamental purpose of this procedure is to avoid risks such as linear cryptanalysis, mathematical assaults, differential attacks, and so on [22], [23]. The S-box lookup static Table is employed to substitute bytes. This is the non-linear process of the approach. This S-box consists of a GF non-linear integral operation (2^8). This S-box is designed to combat assault based on algebraic properties by merging the inverse with an irreversible affine transformation approach. These S-boxes were chosen to avoid static areas [24].

d) Shift Rows: This is the basic linear procedure on state matrices. The method is carried out to achieve diffusion [22], [23]. The rows are subdivided into a few simple stages. For AES, the first row remains unchanged. One place is shifted to the left in each byte of the second row. Two and three places have been changed in the third and fourth rows. As a result, every output block in this phase is made up of bytes from the input block's four columns [24].

e) Mix Columns: This is another essential operation similar to the shift row operation. Matrix multiplication is necessary [22], [23]. This operation set a column from the state array, performs matrix multiplication with the affixed matrix, and generates an output column [21]. In combination with ShiftRows, MixColumns creates the algorithm diffusion properties. As a result, this phase may be seen as a matrix propagation in the final fields [25]. The AES encryption and decryption process are shown in Figure 3.

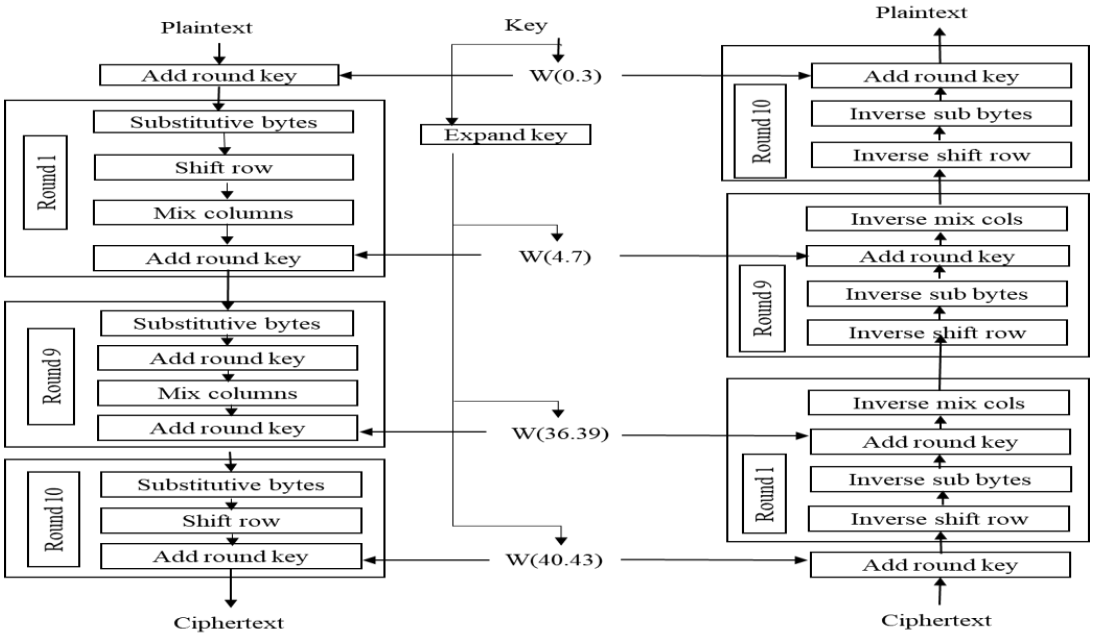

Figure 3. AES Encryption/Decryption process [24]

2.2.2 Data Encryption Standard (DES) Algorithm

The (DES) was created by IBM in1977, it uses a standard technique to protect both sensitive and publicly available information. It divides a data block into two parts, the right half of which passes through a function [23]. DES employs 16 rounds, with S-boxes (Substitution), P-boxes (Permutation), and XOR used in each round. This approach utilizes a 64-bit block size, a 56-bit key size, and 16 rounds with various keys in each round. [26], as shown in Figure 4.
The architecture of the DES algorithm can be summarized as follows:

a) The method of generating keys

The 64-bit key is used as input; 8 bits are utilized as parity check bits remaining during the 56-bit key is split into two halves, with 28 bits on the left and 28 on the right [27]. Depending on the round, every half of the key is moved by one or two bits. Both components have been combined. TheTheyved two sets of 28-bit data are substitutes by selection 2, each group of 24-bit data is processed, and the two filtering sets of data are merged into 48-bits as the sub-key of the i^{th} round [28].

b) First Permutation (IP)

The DES encryption process starts with a permutation operation, termed the IP (Initial Permutation), to modify the order of bits, The DES main function is then run 16 times before being closed with the IP-1 randomizer (Inverse Initial Permutation) [27].

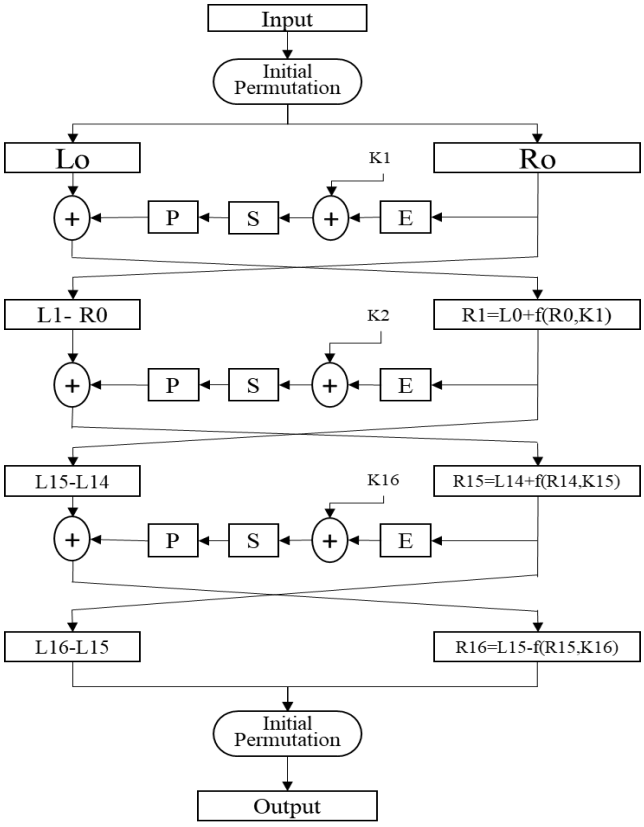

Figure 4. DES algorithm architecture [29]

c) Rounds

In the DES algorithm, there are sixteen rounds. Each round is connected to f(R_{i-1}, K_i). The 64-bit data from the IP block is further divided between the left and right 32-bit in the initial round. In the next round, the current round is left, and the right parts rely on the previous L_{i-1} and R_{i-1} round outputs. This is summarized as:

$$L_i = R_{i-1} \quad (1)$$

$$R_i = L_i \oplus f(R_{i-1}, K_{i)} \text{ [27]} \quad (2)$$

The XOR technique is applied to 48-bit and 48-bit subkeys and then transferred to the S-box. The right half of the input is sent to the expansion E-box, which increases all bits from 32-bit resulting in a 48-bit output. To perform substitution operations, the S-Box has a 5-multiplexer combination circuit. Once the 48-bit data is divided, then sent to eight S-boxes. Each S-Box gets a 6-bit input and provides a 4-bit output via non-linear transformation and a 32-bit combined output via substitution. This is based on a conventional 16-column, four-row Table. The S-box output of 32 bits is delivered directly to the E-box. This computation is carried out 16 times[27].

d) The permutation is finished

The last permutation technique is the inverse of the first permutation function. Encryption gets the ciphertext, while decryption gets the plaintext in this last phase. Thus, the reverse operation is utilized in the encryption and decryption procedures [27].

2.2.3 Triple Data Encryption (TDES) Algorithm

TDES was developed to address the flaws in DES without requiring the creation of a new cryptosystem by IBM in 1978. TDES retains the present investment in equipment and software by employing many encryptions using DES and varied keys. Triple DES enlarges the key size of DES by

running the algorithm 3 times in various keys. It works similarly to DES in that it accepts 64-bit plaintext and outputs 64-bit ciphertext. Unlike DES, however, the total key size is 192 bits, although only 168 bits are utilized (3 times 56). As a result of repeating the DES algorithm three times, TDES is slower than other block cipher algorithms [30].

The operation begins with the DES algorithm encrypting plaintext with key K1, then decrypting the output obtained from the previous step with key K2. The preceding phase's result is then re-encrypted with K3. The ciphertext is the outcome of the final step using k3. Decryption works in the opposite direction to encryption. The encrypted text is then decrypted using the reverse process. K3 is used for decryption, K2 is used for encryption, and K1 is used in the last phase for decryption. Although a TDES-based system is more secure than a DES-based system, it is often slower due to the three-step method [31].

This algorithm specifies three choices of choosing keys from a collection, as shown in Figure 5.

The first choice: provides K1, K2 and K3 are three keys which mutually independent. Which (K1≠ K2≠ K3)." It results in a 3 x 56 = 168-bit ciphertext".

The second choice: applies two mutually exclusive keys and a K3 identical to the first (K1 ≠ K2 and K3 = K1). This is equivalent to a ciphertext of 2 x 56 = 112 bits.

The third choice, A key collection that employs the identical keys (K1 = K2 = K3), is used. The DES technique is the same in both cases [11].

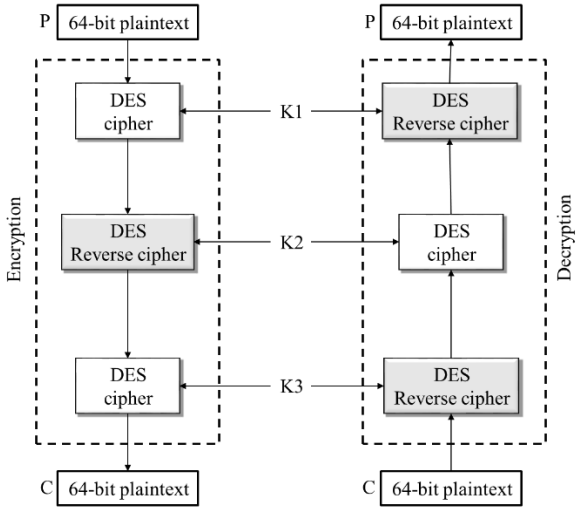

Figure 5. TDES algorithm [32]

2.2.4 Blowfish Algorithm

The Blowfish algorithm was created by Bruce Schneier in 1993, and it has yet to be cracked. It could be better in hardware applications because of its simplicity [32]. It's divided into key expansion and data encryption [33]. Figure 6. Shows how the algorithm works.

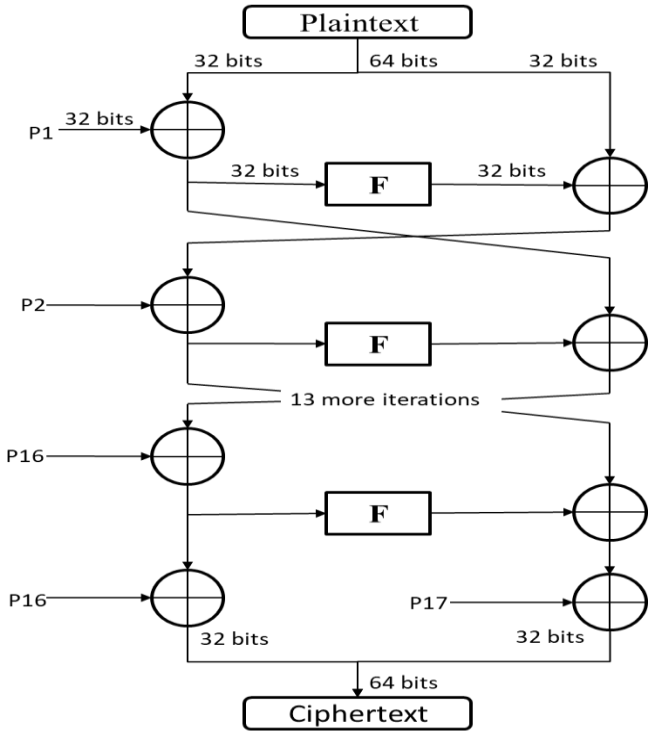

Figure 6. Blowfish algorithm [32]

a*)* Expansion of the keys

The algorithm for blowfish has a minimum of 32, with a maximum function key of 448 bits. The technique of key expansion turns a 448-bit key into several 4168 bytes sub-key arrays. Two sub-key arrays remain in the algorithm: The P-array has 18 32-bit S-boxes and four 32-bit S-boxes, each with 256 elements [34].

The stages in the method below are:

-Subkeys include:

P-array: This has 32-bit subkeys, with a total of 18 identical sub-keys in the array p_1,p_2, $p_3 \ldots p_{18}$.

It also includes key-dependent S-boxes. There are four of them such boxes, each with 256 distinct entries.

─ Initiating the subkeys:

- First, use a static string to set the P-array and 4 S-boxes. This string includes hexadecimal digits of π.
- Then, The first group of subkeys is then XORed with the first p_1 array, the second group of subkeys XORed with the second $p_2$array, and so on until all P- arrays are XORed.
- Based on the above procedure, the string zero is encrypted.
- Step 3 produces the new output order for $p_1$and p_2, then encrypt p_1 and $p_2$again using the new modified subkeys.
- The new order isp_3 andp_4.
- Repeat this process 521 times to produce new subkeys for the P-array and the four S-boxes [35], [36].

b) Data encryption

The 64-bit plaintext block is separated into two parts: left plaintext half (PL) and right plaintext half (PR).

Initially, the data input is XORed with and $p_1$function F. After each output has been completed round; the outputs are switched. To undo the last swap, exchange PL and PR again, then repeat this process until 16 rounds are completed. When all 16 rounds have been finished, the results XORed with p_{17} andp_{18}.

Finally, merge the 32-bit outputs: PL and PR, to obtain 64-bit ciphertext [35].

The function F
In this encryption, the important task is function F.
-The 32-bit entry is divided into four 8-bit quarters in the Blowfish F- function, and each quarter is an input in the S-boxes.
- The modulo 2^{32} is used to add an XOR to produce the final output of 32-bit [35], as shown in Figure 7.

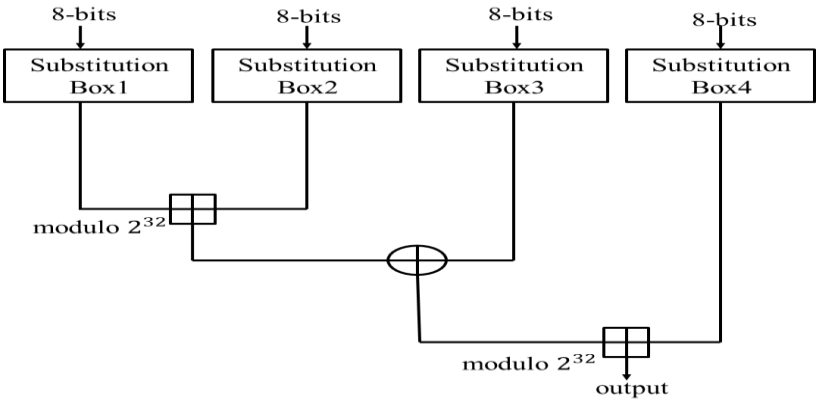

Figure 7 Basic F – Function Feistel Network [35]

2.2.5 Twofish Algorithm

This algorithm was created by Bruce Schneier, John Kesley, and four collaborators in 1998. Two-Fish is a kind of symmetric block encryption. This method utilizes a secret key for both the encryption process and the decryption process. It is one of the algorithms that help secure a data center. The twofish approach secures a 128-bit block size and a 256-bit key length [37]. Two-fish has the following important characteristics:

a) This approach is appropriate for systems with 32-bit and 8-bit processors.
b) It is one of the most secure encryption methods for dealing with brute force attacks.
c) This data encryption technique has been repudiated due to its slower speed when compared to AES-256 [38].

The fundamental feature of the Twofish algorithm is that it has a complex scheme of encryption and pre-calculated S-blocks and is key-dependent. The two components of the n-bit keys are as follows: The encryption keys are utilized half of the time, while the algorithm is changed with the other half [38]. Steps in the Twofish algorithm are shown in Figure 8.

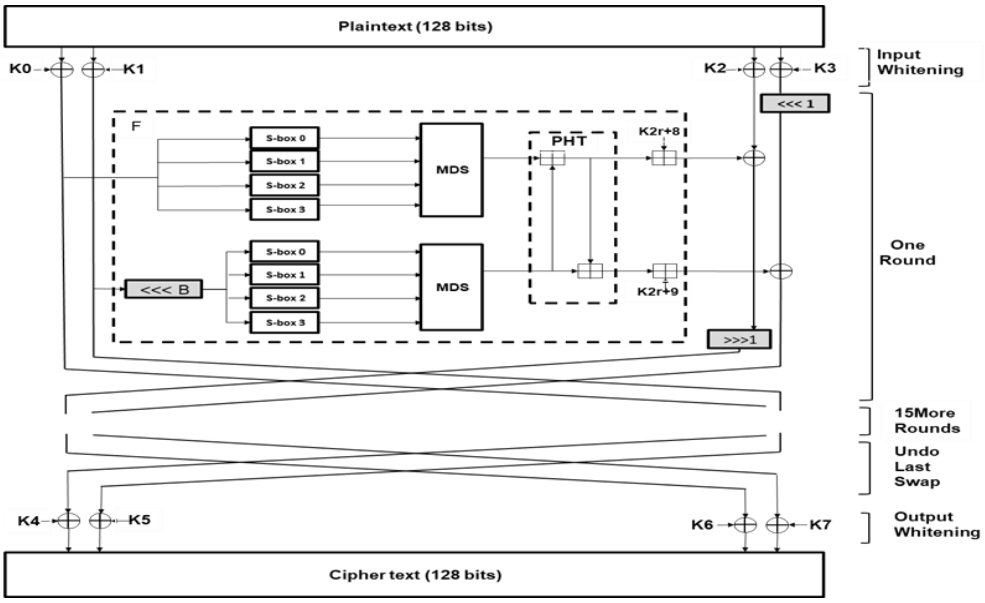

Figure 8 Two fish encryption process [39]

a) S-Boxes

An s-box is a Table-driven non-linear substitution process in block ciphers. The 2 static 8-by-8-bit permutations and key material make up these s-boxes [40]. S-boxes are available in a variety of input and output sizes, and they can be created either randomly or programmatically. S-boxes were initially used in Lucifer, DES, and finally in almost every encryption algorithm.

b) MDS Matrices

A Maximum Distance Separable (MDS) algorithm constructs a composite vector of a+b components with the requirement that in every non-zero vector, The number of non-zero values with the smallest number is less than b+1. The distance between two unique vectors created by MDS mapping is at least$b + 1$. It is simple to demonstrate that no mapping can have a bigger minimum distance between two different vectors; thus, the term "maximum distance separable." An MDS matrix with $a * b$ elements can represent an MDS mapping [40].

c) Whitening

Whitening is the process of XORing key before and after the first cycle. It has been discovered separately that whitening increases the complexity of key search assaults against the remainder of the cipher. Reduced in size round The assault is focused on Twofish variants [41].

d) Scheduling Primary

The key schedule is the procedure for converting key bits into round keys that the cipher may use. The key schedule utilizes the same primitive as the function round to simplify analysis [41]. Twofish has a complicated key schedule and requires a lot of key material**.**

e) Hadamard Transforms (PSEUDO)

Given two inputs, a and b, a pseudo-Hadamard transform (PHT) is a fundamental mixing process that may be done quickly in software. On most modern microprocessors, including the Pentium series, the PHT may be performed in two opcodes[40].

f) Twofish cryptography elements

The input and output data in Twofish are XOR-ed utilizing 8 sequentially (Key0... Key7) subkeys; this is known as input whitening and output whitening. The f function has several processes, including an 8-bit left-shift, S-key-based boxes, MDS, PHT, and 2 sub-keys2^{32}. Function G is also provided, which comprises 4 different S-boxes which all depend on the key. The function G appears twice in the algorithm's structure, generating substantial duplication of the operations associated with the encryption process [42].

g) Functions of Twofish

1) The f Function

The most important part of each Twofish round is the function f. It describes as a key-based permutation of 64-bit values. Vectors R_0 and R_1 are entered together along the round number r. It comprises two parallel G functions, the PHT function and expanded key K. Vector R_0 enters the G function, resulting in vector T_0. Vector R_1 is turned to the left by eight bits before entering the G function and generating a vector T_1. T_0 and T_1 are both vectors that enter the function PHT. Lastly, they are merged with the expanded keys utilizing the XOR processes, producing vectors T_0 and T_1.

$T_0 = \mathrm{g}(R_0)$, $T_1 = \mathrm{g}(\mathrm{ROL}(R_1, 8))$
$F_0 = (\mathrm{T0} + \mathrm{T1} + K_{2r} + 8) \bmod 2^{32}$,
$F_1 = (T_0 + 2T_1 + K_{2r} + 9) \bmod 2^{32}$ [43]

2) The G Function

The G function is made up of four S-boxes. The MDS function is the next operation. A 32-bit word is produced as the output. These 32-bit words are subsequently sent to the PHT function. The keys are XORed with the data blocks, then turned. The network's left and right sides are switched at the end of each round. The sides are not switched in the last round, and just the last key is added. For bit rotation, the inner ROL and ROR functions are employed [43].

3) The h Function

The core of Twofish is the function *h*. Each S-box accepts 8 bits of input and outputs 8 bits; the 4 results are integrated as a four-dimensional vector on GF (2^8) and 4x4 MDS multiplied. The resulting vector is read as a 32-bit word, resulting from *h* [40].

2.2.6 RC2 Algorithm

It was created in 1987 as a block encryption method by Ron Rivest. RC2 is also known as (ARC2). This algorithm was developed to replace the current DES Algorithm [44]. It was considered a possible DES alternative. The purpose of this strategy was to make it simple to implement on 16-bit microprocessors. If the key encryption is done ahead of time on an IBM AT, this approach is twice as fast as DES. The algorithm comprises three sub-algorithms: key expansion, encryption, and decryption. The following interleaved procedure is followed to complete the 18 rounds:

- Perform five rounds of mixing.
- Do a single round of mashing.
- Apply six rounds of mixing.
- Do a single round of mashing.
- Apply five rounds of mixing.

RC2 uses the key-expansion approach which creates a 64-bit (16-bit word) key size increase based on a complicated algorithm that evaluates each bit of the provided not-consistent-length input key, as shown in Figure 9. Each mixing round uses the "mix-up" transformation four times. The mashing round is enhanced with single 16-bit words from the expanded key [45].

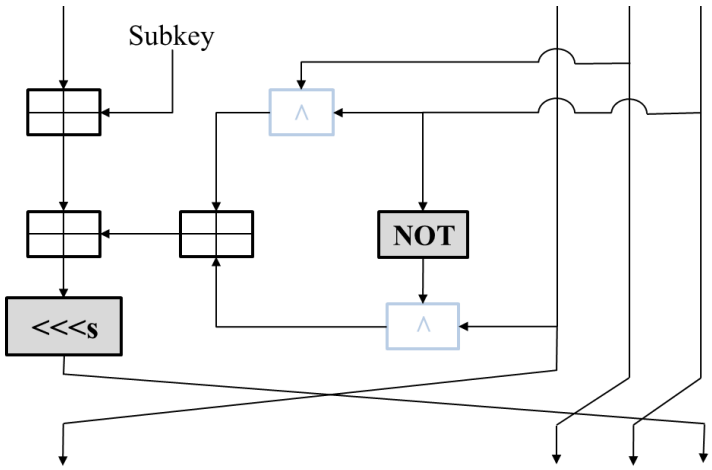

Figure 9 RC2 block cipher [46]

2.3 Asymmetric Encryption Algorithms

Public key techniques use two distinct but related keys for encryption and decryption. The encryption mechanism, as well as the key (public key), are widely known. But decryption key (private key), the owner is the only person who has access to it. It is impossible to extract the private keys from the known public keys in a reasonable time. So, a strong one-way function connects public and private keys [47], as shown in Figure10.

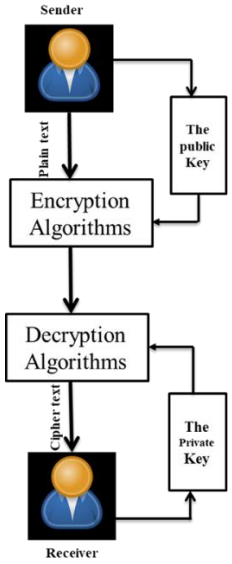

Figure 10. The general idea behind asymmetric-key cryptography [6]

Asymmetric cryptography must meet several essential requirements; 1) the key generation technique should be computationally effective. 2) The sender should compute the ciphertext for each conversation using the receiver's public key. 3) The recipient should rapidly decrypt the encrypted text to plain text using his secret keys. Obtaining the plain text with the public keys and encrypted text is computationally impossible [48].

2.3.1 El-Gamal Cryptography

The El-Gamal algorithm is extremely strong in terms of encryption and decryption and was developed by Taher El- Gamal in 1985. This approach uses the same format to encrypt the private and public key environments. El-Gamal algorithms may be used not only for data encryption but also for digital signatures. The disadvantages of El-major Gamal are his randomness and slower speed (especially for encrypting and decrypting) [6].
The following are the parameters of the Gamal algorithm:

Parameters	Description
q	A separate prime number
x, y	As random numbers
a	Encryption key
y	Decryption key
c	Cipher text
d	Plaint text

The primary calculations of the El-Gamal algorithm can be summarized as follows:

a) Key Pair Generation: The processes for generating a pair of keys for encryption are as below:
1- Choose the random number q as a prime number
2- Select two numbers, x, y as random
Where (x < q) and(y < q)
3- Then compute a=x^y mod q.
4-y is the private key, a is the public key, and the value of x and q also is public [49].
b) The Encryption Equation
Let the message is m; (0 < m < q-1) and choose l as a random number (0 < l < q-1).
Compute c=x^lmod q
d=a^lm mod q

c) The Decryption Equation

The decryption process of c and d is performed using the private-key y and this equation retrieves original text m=$d/c^{l}m$mod q to retrieve the plaintext from the ciphertexts c and d [50].

2.3.2 RSA Algorithm

Rivest - Shamir Adleman developed the RSA encryption technology in 1978. RSA is the most widely utilized algorithm for public key encryption. Although an effective method for factoring huge integers into prime factors has yet to be developed, RSA Algorithm security is still guaranteed. The creators of this approach propose that the prime integer applied to produce the keys be longer than 100 digits. As a result, the multiplication result of that prime value is greater than 200 digits long. Based on Rivest and colleagues, attempting to discover a factor of a 200-digit integer takes 4 billion years of computation time. Prime numbers with more than 100 digits regarded as safe for the RSA method will almost take longer to compute than prime numbers with fewer digits [49].

The calculating processes of the RSA algorithm can be concluded as follows:

The first stage of this encryption process is the creation of RSA Keys. The keys generated utilize the 1024-bit standard RSA. In bidirectional communication, we have public keys and private keys for the transmitter and receiver [10].

The following are the parameters of the RSA algorithm:

Parameters	Description
S and T	two separate prime numbers
A	is the function of Euler's totient
(l, H)	Encryption key
(p, H)	Decryption key
E	Cipher text
D	Plaint text

a) The Generation of Key:

- Choose two prime numbers that are not related to S and T.
- Compute the value of H, which is the multiplication of S and T.
- Calculate Euler's totient function by **α** (H) = (S - 1) x (T - 1)
- Determine the value of l, given these circumstances: 1 < l < **α** (H) where (l, **α** (H)) = 1
- Compute p, where p = l^{-1} (mod **α** (H))

b) Encryption

This equation for encryption

$$E = D^{l} \text{ (mod H)}$$

c) Decryption

The decryption equation

$$D = E^{p} \text{ (mod H)}$$

Where E is the encrypted data, D is the plaintext, (l, H) is the public key, and (p, H) is the private key [51].

2.3.3 Diffie-Hellman Key Exchange (DHKE)

DHKE is a cryptographic key-sharing approach that aims to allow two users to safely exchange keys, It was developed by Whitfield Diffie and Martin Hellman in 1976, it was formerly used to encode and decode data [52]. This key exchange approach enables two unrelated people to create a shared public key across an unsecured internet connection. Key transformations are switched, resulting in the same session key, which seems to be a secret key. Then, each may produce a third key which an adversary who is aware of both exchanged values cannot simply determine. This key is used to encrypt subsequent communications using a key cipher, however, it is vulnerable to Man-in-the-Middle assaults. Unlike

RSA, this key exchange is not utilized for exchanging huge amounts of data [7]. Security depends on the intrusiveness and discreet computational logarithms of the Diffie-Hellman problem [53].
The following are the parameters of the DHKE algorithm:

Parameters	Description
q	a prime number
P	multiplicative group Z_q
A	Secret key for sender
B	Secret key for receiver

The DHKE Protocol between sender and receiver takes place as follows:

Suppose q is the prime number and p is the generator of the multiplicative group $Z_q$that all participants belong to ($2 \leq p \leq q$ -2)

1-The sender is aware of secret a, where $1 \leq a \leq q$ -2 and send:
X=$p^a \bmod q$
2-Likewise, the receiver obtains secret b, where $1 \leq b \leq q$ -2, and sends:
Y=$p^b \bmod q$
3-The sender gets p^b and then calculates
k=$(p^b)^a \bmod q$
4-The receiver gets p^a and then calculates
k= $(p^a)^b \bmod q$ [52], [53].

3. A SUMMARY OF MENTIONED CRYPTOGRAPHIC ALGORITHMS

This section summarizes the main parameters used to distinguish among the different cryptographic methods investigated in this article, as shown in Table (1) [12, 46, 54, 55, 56, 57, 58, 59].

Parameters	AES	DES	TDES	Blowfish	Twofish	RC2	RSA	DHKE	El-gamal
Type	symmetric	symmetric	symmetric	symmetric	symmetric	symmetric	asymmetric	asymmetric	asymmetric
Structure	Substitution - Permutation	Feistel	Feistel	Feistel	Feistel	Feistel	Factorization	Festial & Substitution	discrete logarithm
Key used	Secret key	Secret key	Secret key	Secret key	Secret key	Secret key	Different keys	Different keys	Different keys
Key size	128- 192 and 256 bits	56- bits	192 bits, with only 168 bits being used (3 times 56)	From 32 to 448 bits	128-192and 256 bits	8 to 1024 bits;	1024 to 4096 bits.	1024 to 2048 bits	1024 bits
Round for encryption	10,12 and 14 rounds	16 rounds	48 rounds	16-round	16 rounds	16 (Mixing) +2 (Mashing)	one-round	one-round	one-round
Throughput of encryption and decryption	fast	fast	fast	fast	fast	fast	slow	slow	slow
Battery Consumption	high	medium	high	Very low	low	high	low	high	low

Table 1. A comparison of cryptography methods' common parameters

4. SECURITY IN THE CLOUD- LITERATURE REVIEW

Several recent studies that addressed cloud security have focused on achieving security through encryption techniques, whether by using symmetric, asymmetric algorithms, or hybrid. Ahmed and Garg (2019) have studied the throughput of the AES, RSA, DES, and ECC encryption algorithms to solve the issue of cloud security. This study concluded that, among the mentioned algorithm AES is considered the finest encryption technology due to its speed and flexibility. So, AES is widely used in small devices [60]. In the same context, Mithapalli and Joshi (2019) have studied the performance of AES, Triple DES, and RC2 and compared them. The three encryption and decryption algorithms were used for text and picture data types. Upload and download times, encryption and decryption times, upload and download bandwidth, and encryption differences were used to evaluate the utilized algorithms. The AES technique is the best decision in terms of encryption process time and download bandwidth, according to the results of the trial and comparison. If secrecy and integrity are essential, the AES algorithm is used [61].

Narasingapuram and Ponnavaikko (2020) have proposed another strategy for the stated security issues. They have provided a huge challenge, motivating us to develop a better approach to prevent unauthorized cloud user activity. This study employed DNA cryptography to produce a strong key for the user for data encryption and decryption mechanism. According to the findings, the proposed DNA methodology took less time to computation than the other current ECC and HECC techniques [62]. Another strategy has been proposed by Sohal and Sharma (2018). They have presented BDNA, a new symmetric key encryption scheme.The system obtains the data owner's private key and encrypts the data with it. When users need to access a file, they utilize the framework's interface to provide their login credentials. The security framework requests the user's private key and downloads the appropriate file from the Cloud. The decrypted data is using the DNA decryption method, and the secret key is used to send the file to the user. They have compared their technique to other symmetric-key algorithms and presented the technical architecture of other approaches (Blowfish, DES, AES, and DNA). According to the results of the experiments, the new technique is more efficient and outperforms the standard algorithms in terms of cipher- text size, encryption time, and performance[23].

Furthermore, Thabit et al. (2021) have created the new lightweight symmetric cryptographic Algorithm (NLCA). This algorithm's process employed a block cipher sized128-bitt and a key-sized 128- bit. The technique is built on the Feistel and SP architectural methodologies for increasing encryption difficulty. The suggested method was evaluated with DES, AES, HIGHT, Blowfish, and LED cryptographic algorithms, using a range of key length, security power, cipher type, potential key, block size, and mathematical operations. The results of the experiments revealed that the NLCA technique provides a high level of security and a considerable improvement in encryption and decryption, resulting in high privacy and low computational costs. [63].

Furthermore, Malviya and Dave (2018) have proposed the encryption process carried out on the cryptographic server (CS) portal and provided access permissions (e.g., read, write, and sharing) to a specific user who needs to access protected data. A file can be shared with anybody who has authorization to it once it has been properly encrypted. The data is encrypted using the AES method, while the key is encrypted using the homomorphic (paillier) approach [64].

Gupta et al. (2018) proposed a novel approach to cloud data security. It improves RSA algorithm performance by using a multithreading approach to achieve lower computation overhead. This method can be used to speed up encryption, decryption files, and other RSA implementations such as signature generation and verification. The proposed solution is based primarily on a multithreading technique designed for multi-core CPU systems. They have parallelized the encryption and decryption of numerous data blocks. Initially, the block and key sizes are 100 bytes and 1024 bits, respectively. In order to compare results for various file sizes, the implementation uses both sequential and parallel RSA encryption systems. This study compares different cryptographic methods like RSA, AES, KP-ABE, and

others and discovers that the proposed algorithm offers greater speed and strength than conventional algorithms[65].

Another approach has been suggested in [38] by Jintcharadze and Iavich (2020). They have suggested examples of hybrid Symmetric and Asymmetric cryptosystems such as (Twofish and RSA), (AES and RSA), and (AES and ElGamal). According to the results, the hybridization between AES and RSA is the most secure of the new hybrid models since it incorporates all of the advantages of symmetric and asymmetric systems. However, the combination between Twofish and RSA provided a faster cryptosystem. Liu et al. (2018) have implemented hybrid AES and RSA algorithms for e-mail transmission using the Java programming language. The encryption approach combines the advantages of the AES algorithm's quick encryption speed, the efficiency of managing the RSA algorithm key, and a digital certificate to safeguard private data transfer [66].

According to Abroshan (2021), has suggested an efficient cryptography solution with very little performance impact to increase security in cloud computing. Because computing speed is so important in the cloud computing environment, a complicated cryptography algorithm is not helpful. The original data will be hashed using MD5 in the first step, which will be used to check the accuracy of the data. The second step involves creating digital signatures and protecting the MD5 code and private key using the Elliptic Curve (EC) algorithm. The execution time of the EC will be expedited because they need a high level of security and a short key and hash code. Finally, The original data will then be encrypted with an improved Blowfish algorithm. In comparison to AES, DES, 3DES, and RSA, the solution assessment shows an overall improvement. The proposed solution outperformed the other solutions in terms of throughput, memory usage, and execution time. When we increased the data size, we discovered that AES was slightly faster and had higher throughput [67].

El-Attar et al. (2021) have proposed an (AEDS) System to improve data privacy and confidentiality while avoiding CTP interference for Cloud Storage Systems based on hybrid encryption techniques. To achieve great performance and efficiency, three encryption approaches have been implemented: ASC, ARC, and IARC. The three approaches were compared to other current symmetrical key algorithms such as DES, 3DES, and RC2 using four algorithms; RSA, Twofish, AES, and DES. Although the fact that the two proposed ARC and ASC techniques are more complex, they process data quicker and provide more data throughput and security than DES, DES3, and RC2. In the comparison, ARC beat all of the other methods [68].

Thabit et al. (2022) have proposed a novel, efficient, lightweight homomorphic cryptographic algorithm with two layers of encryption. The first layer uses the new effective, light-weight cryptographic algorithm, and the second layer employs multiplicative homomorphic schemes that are being considered for improving data security in cloud computing. A novel 128-bit lightweight cryptography technique is used in the first layer. Architectural process based on feistal, permutation, and substitution with Shannon's theory of diffusion and confusion, which is based on the involvement of to improve logical operations such as XNOR, XOR, swapping, and shifting. This method incorporates symmetric and asymmetric cryptography features. Memory, computational time, key sensitivity, statistical analysis, image histograms, and entropy change analysis are used to assess the performance of the proposed approach. The experimental results of the proposed algorithm revealed a high level of security as well as an apparent improvement in encryption execution time, memory usage, and throughput. When compared to the standard cryptographic algorithms HIGHT, SEA, LED, RC6, and NLCA widely used in cloud computing [69].

Fatima et al. (2022) have presented a model based on Rivest–Shamir–Adleman, Advanced Encryption Standard, analyze the well-known symmetric algorithm and asymmetric algorithm based on time complexity, space, resource and power consumption. the experimental analysis, The AES algorithm protects data with its high security level and can counterattack against a variety of attacks. The AES is

the method of lesser time complixtey due to it scalable behavior it is easily implemented, leaving the RSA algorithm behind in terms of memory requirements.Unlike RSA, The AES algorithm consumes less storage space while providing high results with no significant limitations. However, as technology advances, hybrid models are replacing traditional security algorithms. Consequently, a hybrid AES/RSA model will enhance cloud security as a whole [70].

Adee and Mouratidis (2002) proposed a data security model based on Rivest-Shamir-Adleman, AES, and identity-based encryption algorithms, as well as Least Significant Bit steganography. By protecting data confidentiality, privacy, and integrity from attackers, this proposed model ensures more cloud redundancy, flexibility, efficiency, and security. To protect cloud data, a dynamic four-step model with hybrid encryption was introduced, which pairs the AES-256 symmetric method with the RSA asymmetric technique. Using the LSB steganography technique, the encrypted data is then hidden in a photograph. The strategies chosen by the users can be used to support the decryption process's results. The results of the encryption and decryption can be shared and securely transferred to authorized recipients using identity-based encryption (IBE). The results also show that the amount of data hidden in the image increases as distortion is reduced. The suggested methodology is more adaptable, flexible, and effective for protecting cloud data for a wide range of businesses with varying sizes, goals, and demands [71].

Table (2) highlights the characteristics of the mentioned research scholars. It shows a summary of the comparison between the research literature. If a column metric is marked, the work in that row addresses it. If it is not marked, then the work either does not specify or does not address that metric. So define these metrics of the table are as follows:

Algorithms used: write all proposed algorithms used in the study.

Algorithms compared: write all algorithms that are used in the comparison with proposed algorithms.

Symmetric/ Asymmetric: informs if the proposed algorithms used in the study were symmetric or asymmetric.

Hybrid approach: if the study has a hybrid between Symmetric and Asymmetric algorithms for encryption;

Environment:

Light: informs if the study tested lightweight ciphers;

Throughput: if the study calculates the amount of data that can be processed in a predefined time;

Time: If the work tested encryption/decryption times or not;

Performance: Performance Analysis of Each Encrypted algorithm according to which algorithm is faster uploading, less time to download files, less time taken to encrypt files, less time is taken to decrypt files, and download bandwidth should be less for optimized performance.

Table2. A comparison of the works mentioned above

Ref.	Used algorithms	Algorithms compared	symmetric	asymmetric	hybrid	Environment	lightweight	Time	Compute throughput	performance
Ahmed and Garg (2019)	DES, AES, ECC, and RSA	DES, AES, ECC, and RSA			√	The proposed algorithms implemented using, an Intel Core i7, 2.7 GHz CPU laptop has been used, in which performance data is computed using CrypTool 2.1 software			√	
Mithapalli and Joshi (2019)	AES, Triple-DES, and RC2	AES, Triple-DES, and RC2	√			(not provided)		√	√	√
Narasinga puram and Ponnavaik ko (2020)	DNA	DES, TDES, Blowfish, ADNA, AES, ECC and HECC	√			DNA implemented in C#.NET language of DOTNET Frameworks over Windows-10		√		
Thabit et al. (2021)	NLCA	DES, TDES, AES, Blowfish, and LED	√			The experimental setting consists of the Xen Server hypervisor as an Open stack middleware and a client that uses Citrix Desktop to access the Xen-Server-hosted virtual machine). The cloud server Details as Core I7 (4.8 GHz) with 8 GB of RAM, and the client computer utilizes the Core I5 with 8 GB RAM.	√			
Sohal and Sharma (2018)	BDNA	AES, DES, Blowfish, and DNA	√			The proposed algorithm has been implemented in java on a core i5 processor 3210 M @ 2.67 GHz, 4 GB RAM, 64-bit operating system	√	√	√	
Malviya and Dave (2018)	Homomor phic and AES		√		√	This approach is implemented by using the JAVA environment and using JSP for web application deployment		√		√
Gupta et al. (2018)	RSA	KP-ABE,CP-ABE and AES		√		This approach is implemented on the Intel-based Corei3-5005U processor with 4CPU cores. by using Java 8 programming language with NetBeans 8.1		√		

						IDE on windows 10 of 64-bit operating system				
Jintcharadze and Iavich (2020)	Twofish, AES, RSA, and ElGamal	AES + RSA, Twofish +RSA and AES+ElGama			√	Using built Java code		√		
Liu et.al. (2018)	AES and RSA	AES and RSA			√	These approaches are implemented by python script language based on the Pycharm development platform		√		
Abroshan (2021)	MD5, EC, and Blowfish	AES, DES, TDES, and RSA			√	Implemented the proposed solution on Eclipse and by using Java development kit (JDK) version 7. The hardware used in the solution evaluation had an Intel Core2 Duo 2.5 GHz processor. All the evaluations were run in Windows 7.		√	√	
Noha E. El-Attar et al. (2021)	RSA, AES, DES, and Twofish	DES, 3DES, and RC2			√	The experiments have been performed using python as a programming language, running on VMware based on a virtual machine, Intel (R) Core i7 2.3 GHz CPU, 32 GB of memory, and Windows 10 operating system	√	√	√	
Thabit et al. (2022)	PROPOSED ALGORITHM	AES, DES and Blowfish RSA, EGAMAL			√	Algorithms have been implemented in MAT.LAB. and applied in C++ using the Dev. C++ program, Omnet C++ v6 was developed to implement a proposed algorithm made in a private cloud Omnet installed on the DELL Inspiron 13 7000 series on an Intel CoreTM i7-3120 M processor, 2.50 GHz 16G.B. RAM.	√	√	√	
Fatima et al. (2022)	AES , RSA	AES , RSA			√	Windows Azure SDK and running application over cloud		√		
Adee and Mouratidis (2002)	AES , RSA and LSB steganography	AES , RSA and LSB			√	The Python programming, Some of the libraries installed using pip		√		

						included NumPy, AES, RSA, PIL, CV, Cryptodome, and Matplotlib				

5. CONCLUSION

Nowadays, it has become essential for organizations to migrate to the Cloud to store their data to save cost and time. However, data security remains a major concern in the cloud environment. With the rising demand for cloud applications, it has become vital to create efficient, resilient, and high-security algorithms appropriate for the large amount of data in the Cloud. In this article, we have provided an overview of some cryptographic algorithms. Some of these algorithms are symmetric, such as TDES, AES, DES, and blowfish, while others, such as RSA, Elliptic Curve, and Diffie-Hellman, are asymmetric. Based on this review, we may conclude the following shortcomings in recent approaches used to maintain data security: All of the research analyzed is based on two or three cryptographic algorithm metrics. The most important factor is the time required for encryption and decryption. These criteria are insufficient to evaluate whether or not the algorithm is the best; also, using a single algorithm (non-hybrid techniques) is insufficient to achieve high levels of security because symmetric algorithms use only one encryption key to encrypt and decrypt data. also some researchers do not calculate additional parameters that are required to increase security, such as throughput and performance, and do not compare all conventional algorithms to determine which is the best.

In future work, we will evaluate the performance of these algorithms on various types of data with varying sizes. In addition, we will execute a hybrid of symmetric and asymmetric algorithms to determine which algorithms are better to enhance the security of cloud computing. On the basis of these studies, we will proposed to test the algorithms' performance using additional performance indicators, such as throughput of encryption and decryption, diffusion analysis, Memory use, power consumption, and CPU process time and the performance.

6. REFERENCES

[1] Kaplancal, U. T., Akyol, M., In Multidisciplinary Digital Publishing Institute Proceedings, 741(1), 1- 11, 2021, doi.org/10.3390/proceedings2021074011.

[2] Bharathi, C., Vijayakumar, V., Pradeep, K.V., Procedia Computer Science, 50, 103-108, 2015, doi:10.1016/j.procs.2015.04.068.

[3] Manikandasaran, S.S., Int. J. of Advanced Research in Computer and Communication Engineering, 5(1), 97-100,2016, doi:10.17148/IJARCCE.2016.5123.

[4] El Makkaoui, K., Ezzati, A., Beni-Hssane, A., Ouhmad, S. , Procedia computer science, 134, 83-90, 2018, doi:10.1016/j.procs.2018.07.147.

[5] Bonde, S. Y., Bhadade, U. S., Computing, Communication, Control and Automation (ICCUBEA), 1-5, 2017, doi: 10.1109/ICCUBEA.2017.8463720.

[6] Mallouli, F., Hellal, A., Saeed, N. S., Alzahrani, F. A., 6th IEEE International Conference on Cyber Security and Cloud Computing (CSCloud), 173-176, 2019, doi:10.1109/CSCloud/EdgeCom.2019.00022.

[7] Bhardwaj, A. , Subrahmanyam, G. V. B , Avasthi, V., Sastry, H., Procedia Computer Science., 85, 535–542, 2016, doi:10.1016/j.procs.2016.05.215.

[8] Islam, S. M. J., Chaudhury, Z. H., Islam, S., Electrical and Computer Engineering (CCECE), IEEE, 1–3, 2019, doi:10.1109/CCECE.2019.8861845.

[9] Chandra, S., Bhattacharyya, S., Paira S., Alam, S. S., Science Engineering and Management Research (ICSEMR), 1-8, 2014, doi:10.1109/icsemr.2014.7043664.

[10] Al-Kadei, F. H. M. S., Mardan H. A., Minas, N. A., Advanced Computing and Communication Systems (ICACCS), 1302-1307, 2020, doi:10.1109/ICACCS48705.2020.9074430.

[11] Yassein, M. B., Aljawarneh, S., Qawasmeh, E., Mardini, W., Khamayseh, Y., Engineering and Technology (ICET), 1-7, 2017, doi:10.1109/ICEngTechnol.2017.8308215.

[12] Ordonez, A. J., Medina, R. P., Gerardo, B. D., IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), 201-205, 2018, doi:10.1109/ISCAIE.2018.8405470.

[13] Haque, M. E., Zobaed, S., Islam, M. U., Areef, F. M., Computer and Information Technology (ICCIT), 1-6, 2018, doi:10.1109/ICCITECHN.2018.8631957.
[14] Alharbi, M. F., Aldosari, F. Soh, B. Alharbi, N. F., Int. J. of Computer Science and Network Security, 21 (9), 1-5, 2021, doi.org/10.22937/IJCSNS.2021.21.9.5.
[15] Qadir, A. M., Varol, N., Digital Forensics and Security (ISDFS), 1-6, 2019, doi:10.1109/ISDFS.2019.8757514.
[16] Srilaya, S., Velampalli, S., Recent Trends in Electronics, Information & Communication Technology (RTEICT), 1264-1270, 2018, doi:10.1109/RTEICT42901.2018.9012536 .
[17] Sallam, S., Beheshti, B.D., TENCON, IEEE, 1784-1789, 2018, doi: 10.1109/TENCON.2018.8650352.
[18] Sunil, J., Suhas H. S., Sumanth, B. K., Santhameena, S., IEEE, Innovation in Technology (INOCON), 1-4, 2020doi:10.1109/inocon50539.2020.9298347.
[19] Rani, K., Sagar, R. K., Telecommunication and Networks (TEL-NET), 1-5, 2017 doi:10.1109/TEL-NET.2017.8343557
[20] Jayant, D., B., Swapnaja, A., U., Subhash, V., P., Kailash, J., K. Sulabha, S. A., Int. J. of Computer Applications, 0975 – 8887, 2015, doi:10.5120/20801-3484.
[21] Chauhan, Y. S., Sasamal, T. N., Communication and Electronics Systems (ICCES), 468-473, 2019, doi:10.1109/ICCES45898.2019.9002543.
[22] Ametepe, A. F., Ahouandjinou, S. A. R. M., Ezin, E. C., IEEE International Smart Cities Conference (ISC2), 93-99, 2019, doi: 10.1109/ISC246665.2019.9071658.
[23] Sohal, M., Sharma, S., J. of King Saud University – Computer and Information Sciences, 1-8, 2018, doi.org/10.1016/j.jksuci.2018.09.024.
[24] Dang, T. N., Vo, H. M., Computer and Communication Systems (ICCCS), 682-686, 2019, doi:10.1109/CCOMS.2019.8821647.
[25] Ruby, A. M., Soliman, S. M., Mostafa, H., Circuits and Systems (ISCAS), 1-5, 2020, doi:10.1109/ISCAS45731.2020.9181276.
[26] Semwal, P., Sharma, M. K., Int. J. of Emerging Trends & Technology in Computer Science (IJETTCS), IEEE, 1–7, 2018, doi:10.1109/ICACCAF.2017.8344738.
[27] Vinay, B. K., Kumar, S., mala, S. P., Deekshitha, S., Parimala, K. E., Electronics, Computing and Communication Technologies (CONECCT), 1-5, 2020, doi:10.1109/CONECCT50063.2020.9198555.
[28] Yihan, W. , Yongzhen, L. , Power Electronics, Computer Applications (ICPECA), 220-223, 2021, doi:10.1109/icpeca51329.2021.9362619.
[29] Kristianti, V. E. , Wibowo, E. P. , Pertiwi, A. , Afandi, H. , Soerowirdjo, B. ,Computer and Information Technology (EIConCIT), 208-211, 2018, doi:10.1109/EIConCIT.2018.8878519.
[30] Kansal, S., Mittal, M., Parallel, Distributed and Grid Computing, 105-109, 2014, doi:10.1109/pdgc.2014.7030724.
[31] Vennela, G. S. , Varun, N. V. , Neelima, N., Priya , L. S., Yeswanth, J. , Inventive Communication and Computational Technologies (ICICCT), 273-279, 2018 doi:10.1109/ICICCT.2018.8473148.
[32] Shetty, V. S., Anusha, R. ,Kumar, M.J. D. , Hegde, N. P., Inventive Computation Technologies (ICICT), 167-174, 2020, doi:10.1109/ICICT48043.2020.9112491.
[33] Nie, T., Zhang, T., TENCON, IEEE, 1-4, 2009, doi:10.1109/TENCON.2009.5396115.
[34] Nalawade, S. B., Gawali, D. H., Innovations in Signal processing and Embedded Systems (RISE), 479-484, 2017, doi:10.1109/RISE.2017.8378204.
[35] Anusha, R., Dileep, K. M. J., Shetty, V. S., Prajwal, H. N., Electronics, Communication and Aerospace Technology (ICECA), 765-769, 2020, doi:10.1109/iceca49313.2020.9297547.
[36] Parvathy, P., Remya, A. A. S., Communication and Signal Processing (ICCSP), 0770-0774, 2020, doi:10.1109/ICCSP48568.2020.9182088.
[37] Joshi, M., Joshi, M., Intelligent Engineering and Management (ICIEM), 272-276, 2020, doi:10.1109/ICIEM48762.2020.9160185.
[38] Jintcharadze, E., Iavich, M., IEEE East-West Design & Test Symposium (EWDTS), 1-5, 2020, doi:10.1109/ewdts50664.2020.9224901.

[39] Vatsala, V., Poongodi,T., Int. J. of Recent Technology and Engineering (IJRTE),1906-1910, 2020, doi:10.35940/ijrte.F7295.059120.
[40] Sawant, A. G., Nitnaware, V.N., Dengale, P., Garud, S., Gandewar, A., J. of Emerging Technologies and Innovative Research (JETIR) 6, 2019.
[41] Aparna, V.S., Rajan, A., Jairaj, I., Nandita, B., Madhusoodananand, P., Remya, A.A., Trends in Electronics and Informatics (ICOEI), 1279-1283, 2019, doi:10.1109/ICOEI.2019.8862703.
[42] Hoomod, H. K., Hussein, A. M., Intelligent Computing and Control Systems (ICCS), 1189-1195, 2019, doi:10.1109/ICCS45141.2019.9065573.
[43] Smekal, D., Hajny, J., Martinasek, Z., Telecommunications and Signal Processing (TSP), 1-5, 2018, doi:10.1109/TSP.2018.8441386.
[44] Charbathia, S., Sharma, S., Int. J. of Information & Computation Technology, 1831-1838, 2014.
[45] Gonsai,A.M., Lakshadeep M. R. , Int. J. of Computer Trends and Technology (IJCTT) ,11 (1), 7-12, 2014, doi:10.14445/22312803/IJCTT-V11P102.
[46] Al-Shabi, M. A., Int. J. of Scientific and Research Publications, 9, 2019, doi:10.29322/IJSRP.9.03.2019.8779.
[47] Fanfara, P., Danková, E., Dufala, M., Applied Machine Intelligence and Informatics (SAMI), 213-217, 2021, doi:10.1109/sami.2012.6208959.
[48] Hasib, A. A., Haque, A. A. M. M. , Convergence and Hybrid Information Technology, 505-510, 2008, doi:10.1109/iccit.2008.179.
[49] Iswari, N. M. S. Information Technology and Electrical Engineering (ICITEE), 1-5, 2016, doi:10.1109/iciteed.2016.7863255.
[50] Dissanayake, W.D.M.G.M., Int. J. of Computer Applications Technology and Research, 7, 40-44, 2018, doi:10.7753/IJCATR0702.1002.
[51] Mittal, S., Arora, S., Jain, R., Information Processing (IICIP), IEEE, 1-5, 2017, doi:10.1109/iicip.2016.7975347.
[52] Yusfrizal, Y., Meizar, A., Kurniawan, H., Agustin, F., Cyber and IT Service Management (CITSM), 1-6, 2018, doi:10.1109/CITSM.2018.8674278.
[53] Alias, Y. F., Hashim, H., Computer Applications & Industrial Electronics (ISCAIE), 212-216, 2018, doi:10.1109/ISCAIE.2018.8405472.
[54] Singh, G., Supriya, Int. J. of Computer Applications (0975 – 8887) 67(19), 33-38, 2013, doi:10.5120/11507-7224.
[55] Kubadia, A., Idnani, D., Jain, Y., Computing Methodologies and Communication (ICCMC), 118-123, 2019, doi:10.1109/ICCMC.2019.8819729.
[56] Mehibel, N., Hamadouche, M., Electrical Engineering - Boumerdes (ICEE-B), 1-6, 2017, doi:10.1109/ICEE-B.2017.8192159.
[57] Sharma,N.A., Farik,M., , Int. j.of scientific & technology research, 6(7),292-294, 2017.
[58] Mankotia,S., Sood, M., Int. J. of Computer Science and Information Technologies, 6(1), 495-499, 2015.
[59] Francis, N., Monoth, T., Int. J. of Applied Engineering Research, 13(3), 2018.
[60] Ahmed, Q. W., Garg, S., I-SMAC (IoT in Social, Mobile, Analytics ,and Cloud) (I-SMAC), 205-210, doi:10.1109/I-SMAC47947.2019.9032581.
[61] Mithapalli , A. V., Joshi, S. S., Int. J. of Engineering and Advanced Technology (IJEAT),9, 2019, doi: 10.35940/ijeat.B3794.129219.
[62] Narasingapuram, P. B., Ponnavaikko, M., Int. J. of Recent Technology and Engineering (IJRTE), 8, 3738-374, 2020, doi:10.35940/ijrte.B2845.018520.
[63] Thabit, F., Alhomdy, S., Al-Ahdal, A. H.A., Global Transitions Proceedings, ELSEVIER, 91–99, 2021, doi:10.1016/j.gltp.2021.01.013.
[64] Malviya, S., Dave, S., Int. J. of Applied Engineering Research, 14799-14805, 2018.
[65] Gupta, P., Kumar, V. D., Singh, A. K., Cloud Computing, Data Science & Engineering(Confluence), 14-15, 2018, doi:10.1109/CONFLUENCE.2018.8442788.
[66] Liu, Y., Gong, W., Fan, W., Computer and Information Science (ICIS), 701-703, 2018, doi:10.1109/icis.2018.8466380.
[67] Abroshan, H., Int. J. of Advanced Computer Science and Applications, 12(6), 2021, doi:10.14569/IJACSA.2021.0120604.

[68] El-Attar, N. E., El-Morshedy, D. S., Awad, W. A. Cryptography, MPDI, 5, 37. 2021, doi.org/10.3390/cryptography5040037.
[69] Thabit, F., Can., O. , Alhomdy, S., Al-Gaphari., G. H. and Jagtap, S., International Journal of Intelligent Networks international, 16-30, 2022, doi.org/10.1016/j.ijin.2022.04.001.
[70] Fatima, S. , Rehman, T., Fatima, M., Khan, S. and Ali, M. A., Engerieer Proceding, MPDI, 1-6, 2022, doi.org/10.3390/engproc2022020014.
[71] Adee, R., and Mouratidis, H., sensors, MPDI, 1-23, 2022, doi.org/10.3390/s22031109.