A Comparative Analysis of Symmetric Algorithms in Cloud Computing: A Survey

Enas Elgeldawi
Computer Science Department
Faculty of Science
Minia University, Egypt

Maha Mahrous
Computer Science Department
Minia University
Minia, Egypt

Awny Sayed
Computer Science Department
Faculty of Science
Minia University, Egypt

ABSTRACT
Introducing Cloud computing to the globe has changed many conceptual and infrastructural bases for today's and tomorrow's computing. It has made the global thinking migrates rapidly towards cloud based architecture. Clouds bring out a variety of benefits including computing resources configurability, cost controllability, sustainability, mobility and service flexibility. However, the new concepts that clouds introduce such as outsourcing, multi-tenancy, and resource sharing create new challenges and raise a broad range of security and privacy issues. Cryptography is the art-of-science of protecting data privacy by converting it to unreadable format using standard mathematical techniques. This paper provides a comprehensive study for eight of the most common symmetric cryptographic algorithms, namely, DES, 3DES, Blowfish, Twofish, RC2, RC5, RC6 and AES. A comparative analysis based on the structure of the algorithm, encryption and decryption times, throughput and memory utilization has been performed to examine the performance of each algorithm.

General Terms
Cloud Computing, Cryptography, Symmetric Algorithm

Keywords
Cloud Computing, DES, 3DES, Blowfish, Twofish, RC2, RC5, RC6, AES

1. INTRODUCTION
Cloud computing is the delivery of computing services over the Internet. When you share your latest news with your friends over Facebook, or upload your photos to Instagram, you are using "Cloud Computing". Instead of storing data on your local system or downloading applications you need, you use a service over the Internet at another location, to store your information or use its applications. The cloud computing model allows individuals and organizations to use software and hardware that are managed by a third party at remote locations to access information and to use a shared pool of resources.

Cloud Computing has been defined by the National Institute of Standards and Technology (NIST) [1] as: "Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction." With the introduction of numerous cloud based services, the possibility that sensitive information stored in remote servers could be exposed to unwanted parties is compromised. If security is not consistent and robust enough, the advantages that cloud computing has to offer will have questionable credibility.Cryptography is the key concept used to handle data security concerns in cloud computing. Cryptography is to convert data into unreadable form during storage and transmission. The unreadable form of data is known as cipher text. When data is received, the cipher text is converted back into its original form which is known as plain text. Conversion of plain text to cipher text is known as encryption and the reverse process is known as decryption. Encryption takes place at sender's end whereas decryption takes place at receiver's end. In section 2, we frame the way cryptographic algorithms is classified. Section 3, explains Feistel network which is the major scheme most symmetric ciphers are based on. A detailed study for each algorithm is given in Section 4. Computational analysis is discussed in Section 5. And finally the conclusion key notes are presented in Section 6.

2. CLASSIFICATION OF CRYPTOGRAPHIC ALGORITHMS
Cryptography is the art of science of using mathematical techniques to convert plain text (P) into an unreadable cipher text (C) according to some encryption algorithm that uses a numeric or alpha numeric parameter called key (K1). and similarly, reconvert the cipher text back to plain text via a decryption algorithm which uses a decryption key (K2). This can be interpreted as Cipher text C = E (P, K1) and Plain text P = D (C, K2). This is indicated by Figure 1.

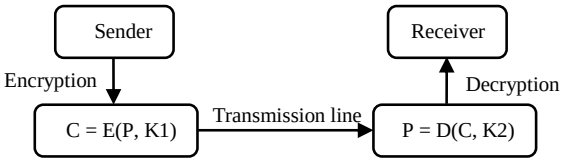

Figure 1: Encryption and Decryption Processes

There are many Cryptographic algorithms in use for cloud computing, they can be classified as follows:

2.1 Symmetric Algorithms
Symmetric Algorithms (also known as secret key algorithm) use a single key for both encryption and decryption, this key is known to both the sender and receiver. Symmetric algorithms can be subdivided into: Block Cipher and Stream Cipher. The major difference between block cipher and stream cipher is that block cipher takes a message and break it into a fixed size of blocks and converts one block of the message at a time [2]. On the other hand, a stream cipher algorithm typically encrypts and decrypts the text by converting one byte of the text at a time.

2.2 Asymmetric Algorithms

In asymmetric algorithms, encryption and decryption are performed using two different but mathematically related keys. a public key and a private key. The public key may be freely distributed, while its paired private key must remain secret. The public key is typically used for encryption, while the private or secret key is used for decryption.

Figure 2 shows the classification of cryptographic algorithms along with examples of each classification and the year they are created.

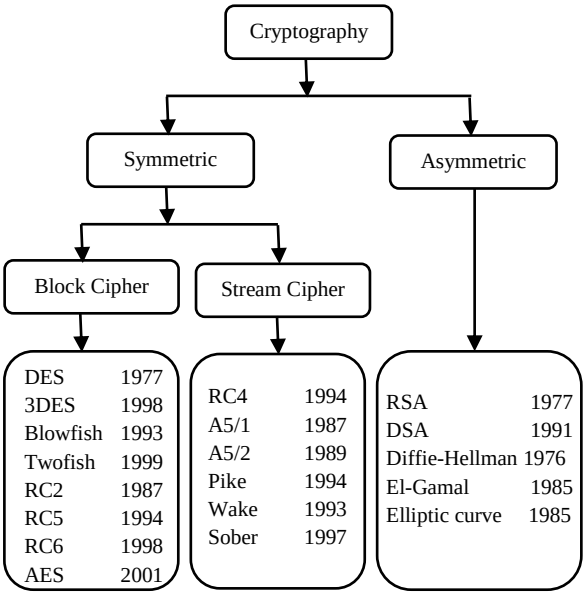

Figure 2: Cryptographic Algorithms Classification

3. FEISTEL BLOCK CIPHER

Most of the major symmetric ciphers are based on a cipher known as the Feistel block cipher or Feistel network. This was a block cipher developed by the IBM cryptography researcher Horst Feistel in the early 70's [3]. It is considered as a general approach to build a block cipher. Such a network divides the plain text into fixed size blocks then takes a block of the plaintext and the key as inputs, and applies several iterations called rounds to generate the cipher text. Each round applies a function (F) that consists of two main operations, substitution (or S-box) and permutation (or P-box). Substitution is a mapping of one value to another whereas permutation is a bit-shuffling process or simply a reordering of the bit positions for each of the inputs. The block size, key size, the function F and number of rounds vary from one cipher algorithm to another. Generally, the more rounds there are, and the larger the key size is, the more secure a cipher algorithm will be. Most symmetric encryption schemes today are based on this structure.

The Feistel structure starts by splitting a block of plain text into two equal right (R) and left (L) halves. The function (F) takes the right half and a sub key from the original master key as inputs, the output of F is then XORed with the left half (L). Then the two sides are swapped to start another round. The structure of Feistel algorithm is shown in Figure 3 and applies the following steps.

1. The plain text is split into two halves L_0 and R_0.
2. For each round $i = 1, 2, \ldots, n$

a. The function F is applied to the right half R and ith sub key yielding $F(R_i, K_i)$

b. The XOR of $F(R_i, K_i)$ and the left half L is computed yielding $L_i \oplus F(R_i, K_i)$

c. The two sides are swapped to set the R and L for the next round, that is $L_{i+1} = Ri$ and $R_{i+1} = L_i \oplus F(R_i, K_i)$

L_{n+1} and R_{n+1} are recombined to construct the cipher text.

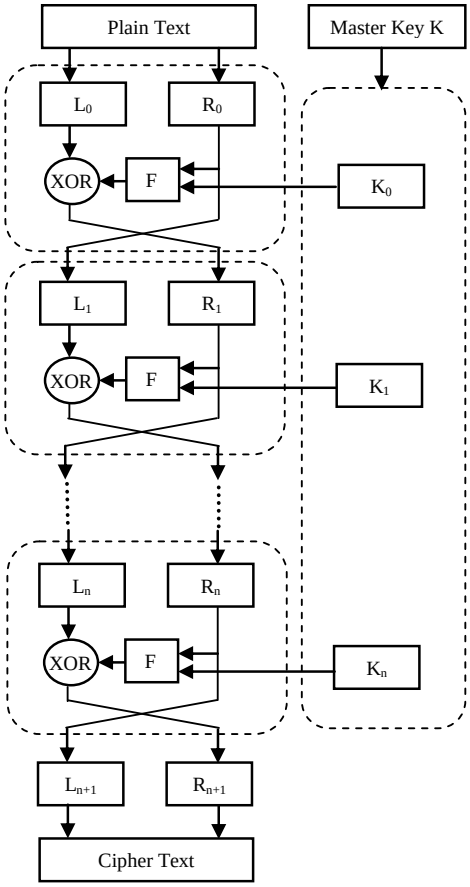

Figure 3: General Structure of Feistel network

4. SYMMETRIC ALGORITHMS ON THE SPOT

As mentioned in the previous section, most of the major symmetric ciphers are based on Feistel block. In this section we review the major symmetric algorithms in used today, namely, DES, 3DES, Blowfish, Towfish, RC2, RC5, RC6, AES.

4.1 DES

The Data Encryption Standard (DES) is one of the most widely used and publicly available cryptographic algorithms. It was developed by IBM in the 1977 and it was the initial encryption standard to be recommended by the National Institute of Standards and Technology (NIST) [4]. DES is a block Cipher which is designed to encrypt and decrypt blocks of data consisting of 64 bits by using a 64-bit key. Although the input key for DES is 64 bits long, the actual key used is only 56 bits in length. The least significant bit in each byte is a parity bit. These parity bits are ignored, so only the seven most significant bits of each byte are used, resulting in a key length of 56 bits. The algorithm goes through 16 Feistel rounds. The algorithm transforms 64-bit input in a series of steps into a 64-bit output. The same steps, with the same key are used for decryption. Many attacks recorded the

weaknesses of DES, which made it an insecure block cipher. Despite the growing concerns about its vulnerability, DES is still widely used by financial services and other industries worldwide for data protection.

4.2 3DES

3DES or the Triple Data Encryption Algorithm (TDEA) was developed by IBM in 1998-1999 to address the obvious flaws in DES without designing a whole new cryptosystem. The 56-bit key used in DES is not deemed secure enough to encrypt sensitive data. 3DES simply extends the key size of DES by applying the algorithm three times in succession with three different keys. The combined key size is thus 168 bits. TDEA involves using three 64-bit keys (K1, K2, K3) in Encrypt-Decrypt- Encrypt (EDE) fashion, that is, the plain text is encrypted with K1, then decrypted with K2, and then encrypted again with K3 [5]. The encryption process can be expressed as *C = E(D(E(P, K1), K2), K3)*, similarly, decryption process can be interpreted as *P = D(E(D(C, K3), K2), K1)*

The standards define three keying options:

- Option 1: the strongest and preferred option, employs three mutually independent key, with 3 × 56 = 168 independent key bits.
- Option 2: K1 and K2 are independent, and K3 = K1, this provides a shorter key length of 2 × 56 = 112 bits and a reasonable compromise between DES and Keying option 1, but NIST has deprecated this option.
- Option 3: All three keys are identical. This option is equivalent to DES Algorithm and NIST no longer allows this option.

While the 3-times iteration is applied to increase security, it is tripled the processing time as well, making 3DES one of the slowest block cipher methods.

4.3 BLOWFISH

Blowfish was designed in 1993 by Bruce Schneier as a fast, free patent alternative to existing encryption algorithms at that time. Blowfish provides a good encryption rate and no cryptanalysis of it has been recorded to date. The algorithm consists of two parts: a key-expansion part and a data-encryption part. Blowfish uses a large number of subkeys which are constructed during the Key expansion phase. This phase converts a variable-length key of at most 56 bytes (448 bits) into an array of sub-keys called P-array consists of eighteen 32-bit sub-keys totaling 4168 bytes. There are also four 32-bit S-boxes. The keys must be computed before any data encryption or decryption [6]. The creation of sub-keys further increases security, because a hacker would have to crack more than just the original key.

The second part of the Blowfish routine that is a data encryption is done through 16 Fiestel network rounds, a swap operation and two exclusive-or operations. The F function takes the 32-bit input and divides it into 4 bytes (8-bits each). These four values are then used for table lookup in their corresponding S-Boxes. A graphic representation of Blowfish is given in Figure 4.

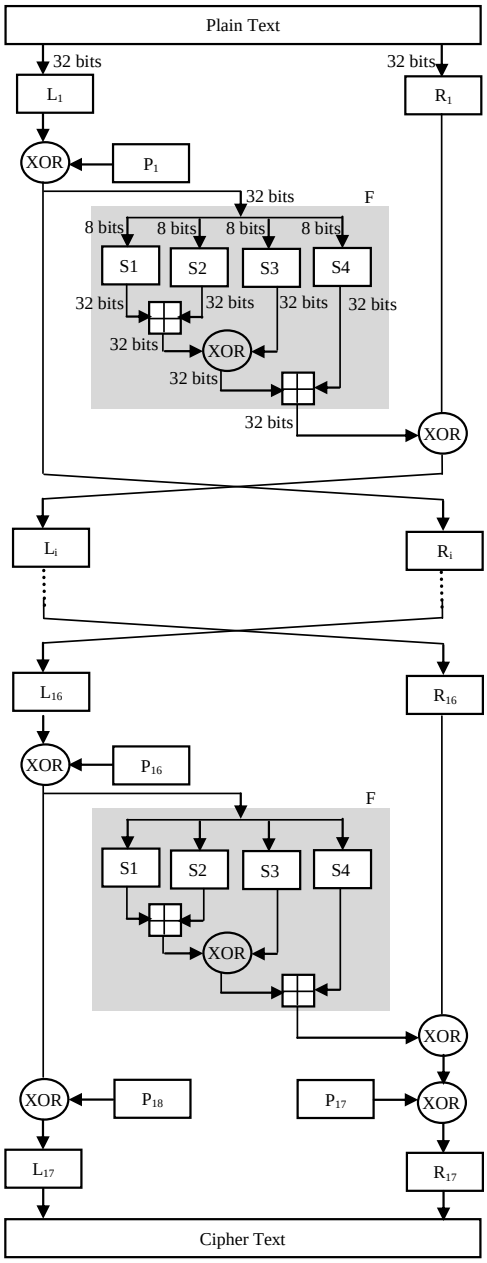

Figure 4: Blowfish Algorithm

Blowfish algorithm is as follows:

1. The 64-bit plain text is split into two halves L_0 and R_0.
2. For each round *i = 1, 2, …, 16*
   a. The left half *L* is XORed with the *i*th sub key $L_i = L_i \oplus P_i$
   b. The XOR of $F(R_i, K_i)$ and the left half *L* is computed yielding $R_i = F(L_i) \oplus R_i$
   c. The two sides are swapped to set the *R* and *L* for the next round, that is $L_{i+1} = Ri$ and $R_{i+1} = L_i \oplus F(R_i, K_i)$
3. Swapp R_{16} and L_{16} again to undo the last swap.
4. $R_{17} = R_{17} \oplus P_{17}$, $L_{17} = L_{17} \oplus P_{18}$
5. *L* and *R* are recombined to construct the cipher text.

The *F* function looks like this: Divide *L* into four eight-bit quarters: *a*, *b*, *c*, and *d*. Then, $F(L) = (((((S + S2) \bmod 2^{32}) XOR S3) + S4) \bmod 2^{32})$.

The complexity of the Blowfish key generating process made it a considerable robust block cipher algorithm [7].

4.4 TWOFISH

Twofish is a symmetric block cipher with block size of 128 bits, and accepts a key of any length up to 256 bits. Twofish is fast on both 32-bit and 8-bit CPUs. It can be used in network applications where keys are changed frequently and where little or no RAM and ROM available. Twofish is a Feistel network [8]. This means that in each round, half of the text block is sent through an F function, and then XORed with the other half of the text block.

In each round of Twofish, two 32-bit words serve as input into the F function. Each word is broken up into four bytes. Those four bytes are sent through four different key-dependent S-boxes. The four output bytes (the S-boxes have 8-bit input and output) are combined using a Maximum Distance Separable (MDS) matrix and combined into a 32-bit word. Then the two 32-bit words are combined using a Pseudo-Hadamard Transform (PHT), added to two round subkeys, then XORed with the right half of the text. There are also two 1-bit rotations going on, one before and one after the XOR. Twofish also has something called "Prewhitening" and "Postwhitening;" additional subkeys are XORed into the text block both before the first round and after the last round.

4.5 RC2

RC2 is a 64-bit block cipher with variable key size designed by Ron Rivest in 1987. "RC" stands for "Ron's Code" or "Rivest Cipher". The plaintext is divided into four words each of 16 bits. There is a key expansion process before the encryption or decryption take place. The key expansion process takes a user key between 1 and 128 bytes in length together with a parameter that specifies the effective key-length of encryption, then an array K of 64 sub keys is derived (16-bit each). Then encryption is done through two kinds of rounds. One is termed a MIXING round and the other a MASHING round. There are in total 16 mixing rounds and two mashing rounds. In each round each of the four words in the intermediate cipher texts is updated as a function of the other words. Each of the mixing rounds takes a 16-bit subkey.

An array of four 16-bit words R[0], ..., R[3] is used to hold the initial plaintext, the intermediate results, and the final cipher text.

One MIXING round is defined as follows, for each i = 0, 1, 2, and 3:

1. $R[i] = R[i] + K[j] + (R[i-1] \,\&\, R[i-2]) + (\sim R[i-1] \,\&\, R[i-3]);$

Where & denotes bitwise logical and, $\oplus$ denotes bitwise XOR and $\sim$ denotes bitwise complementation. All 16-bit word addition + is performed modulo 216.

2. $j = j + 1;$

Here *j* is a "global" variable so that *K[j]* is always the first key word in the expanded key which has not yet been used in a MIXING operation.

3. $R[i] = R[i] <<< s[i];$

   where s[0] = 1, s[1] = 2, s[2] = 3, and s[3] = 5. And R[i] <<< s[i] denote that R[i] is rotated left by *s[i]* bit.

One MASHING round is defined as follows, for each *i* = 0, 1, 2, and 3:

$$R[i] = R[i] + K[\, R[i-1] \,\&\, 63\,];$$

The entire encryption operation with RC2 can now be described as follows [8]:

1. Initialize words *R[0], ..., R[3]* to contain the 64-bit plaintext block.
2. Expand the key, so that words *K[0], ..., K[63]* become defined.
3. Initialize j to zero.
4. Perform five MIXING rounds.
5. Perform one MASHING round.
6. Perform six MIXING rounds.
7. Perform one MASHING round.
8. Perform five MIXING rounds.
9. The ciphertext is *R[0], ..., R[3]*.

Although no design weaknesses have been identified in RC2 which could lead to practical cryptanalytic attacks, it is considered to be a slow cipher.

4.6 RC5

RC5 is a 32/64/128-bit block cipher developed in 1994. It was designed also by Ronald Rivest in December of 1994. It is a symmetric block cipher having a variable number of rounds, word size and a secret key.

As in RC2 there is a key-expansion routine that expands the user's key K to fill an expanded key array S. It is assumed that the input block is given in two w-bit registers A and B, and that key-expansion has already been performed, so that the array S[0...t - 1] has been computed. Here is the encryption algorithm in pseudo-code [9]:

1. $A = A + S[0];$
2. $B = B + S[1];$
3. For *i* = l to *r* do
   a. $A = ((A \mid B) <<< B) + S[2 * i];$
   b. $B = ((B \mid A) <<< A) + S[2 * i + 1];$

The output is in the registers A and B.

It is a simple algorithm which has a low memory requirement. It is fast and yet secure if suitable parameters are chosen.

4.7 RC6

It was one of the AES finalist developed also by Ronald Rivest in 1997. It uses 128 bit block size and supports key sizes of 128, 192 and 256 bits. It is an improvement of the RC5 Algorithm and provides even better security. It makes use of 4 registers (Each one of 32 bit). It uses fewer rounds and offers a higher throughput. The following is the encryption algorithm in pseudo-code, where S is the array of keys generated by the key expansion routine:

1. *B = B + S[0];*
2. *D = D + S[1];*
3. *for i* = 1 to *r* do
   a. *t = (B*(2B + 1)) <<< lg w*
   b. *u = (D*(2D + 1)) <<< lg w*
   c. *A = ((A ⊕ t) <<< u) + S[2i]*
   d. *C = ((C ⊕ u) <<< t) + S[2i + 1]*
   e. *(A, B, C, D) = (B, C, D, A)*
4. *A = A + S[2r + 2];*
5. *C = C + S[2r + 3];*

4.8 AES

The Advanced Encryption Standard (AES), also known by its original name Rijndael, is the encryption standard recommended by NIST in 2001. AES is developed by tow Belgain cryptographers, Vincent Rijmen and Joan Daemen, who submitted a proposal to NIST during the AES selection process. Unlike the other cipher algorithms discussed in this paper, AES does not use a Feistel network. Instead AES is a variant of Rijndael ciphers which has a fixed block size of 128 bits, and a key size of 128, 192, or 256 bits. Rijndael is specified with block and key sizes that may be any multiple of 32 bits, with a minimum of 128 and a maximum of 256 bits. AES algorithm can support any combination of data (128 bits) and key length of 128, 192, and 256 bits. The key size used for an AES specifies the number of rounds needed to encrypt or decrypt the text. AES goes through 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys in order to deliver final cipher text or to retrieve the original plain text. AES allows a 128 bit data length that can be divided into four basic operational blocks. These blocks are treated as array of bytes and organized as a matrix of the order of 4×4 that is called the state. For both encryption and decryption, the cipher begins with an AddRoundKey stage. However, before reaching the final round, this output goes through nine main rounds, during each of those rounds four transformations are performed; sub-bytes, shift-rows, mix-columns, and add round Key. In the final (10th) round, there is no mix-column transformation [2]. Decryption is the reverse process of encryption and using inverse functions: Inverse Substitute Bytes, Inverse Shift Rows and Inverse Mix Columns.

The date block goes through four steps of transformation at each round [10, 11, 12, 13]:

1) Substitute Byte transformation: AES contains 128 bit data block, which means that each of the data blocks has 16 bytes. In sub-byte transformation, each byte (8-bit) of a data block is transformed into another block using an 8-bit substitution box which is known as Rijndael S-box.
2) Shift Rows transformation: It is simple byte transpositions, the bytes in the last three rows of the state, depending on the row location, are cyclically shifted. For 2nd row, 1 byte circular left shift is performed. For the 3rd and 4th row 2-byte and 3-byte left circular left shifts are performed respectively.
3) Mixcolumns transformation: This round is equivalent to a matrix multiplication of each Column of the states. A fix matrix is multiplied to each column vector.
4) Addroundkey transformation: It is a bitwise XOR between the 128 bits of current state and 128 bits of the round key. This transformation is its own inverse

According to many literature reviews [14, 15, 16, 17, 18], AES is considered one of the most secure algorithms. In theory it has not broken since the time of its development.

5. COMPUTATIONAL RESULTS

The main goal of any encryption algorithm is security. However, other performance metrics may determine the preference use of one algorithm over another. In this section, we compare the algorithms on the basis of different parameters, the structure of the algorithm, encryption and decryption times, throughput and memory usage. The experimental results are depicted against varied file sizes and recording the computation cost for those algorithms.

5.1 The Structure of the Algorithm

The structure is characterized by a set of variables such as the block size, key size, and number of rounds. Ultimately, these are the factors which affect the security of a particular algorithm. The key size plays a vital role in algorithm security. Larger key size provides higher security when other factors were considered to be equal in some algorithms. Another critical factor is the number of rounds used for encryption/decryption process. Performing more rounds, strengthens the security of the algorithm, but increases the complexity as well. That is why, when designing a cryptographic algorithm, the number of rounds is one important factor that should be set carefully. Table 1 summarizes this comparison.

5.2 B. Encryption and Decryption Tim

The time needed by every algorithm to encrypt and decrypt varied size files is calculated and analyzed. Table 2 and Table 3 are graphically represented by Figure 5 and Figure 6, they give the results of time in milliseconds of encryption and decryption processes, respectively. Encryption and decryption rates are dependent on the processor speed, and algorithm structure. From the tabular results, it is easy to observe that AES, RC6 and RC2 have the best encryption and decryption performance over other algorithms in terms of time consumption. On the other hand, the complexity of 3DES is clearly reflected on the time it takes to encrypt and decrypt files.

5.3 Throughput

The throughput of an algorithm is calculated by dividing the total block size in Megabyte encrypted or decrypted on the total encryption or decryption time. The throughput of the encryption scheme is calculated using the following formula:

Throughput = Total Plain Text / Encryption Time

Similarly, the decryption scheme is calculated using a similar formula:

Throughput = Total Plain Text / Decryption Time

The throughput of encryption and decryption processes is given in Figure 7 and Figure 8, respectively. The figures show the superiority of Twofish and Blowfish over the other algorithms.

5.4 Memory Utilization

Memory utilization or memory usage is another important parameter that should be taken into consideration. The cost of memory often exceeds the cost of CPU. As a result the efficient utilization of memory has received much attention when designing an algorithm. The memory requirement depends on the key size, initialization vectors, and type of operations. Table 4 and Table 5 give memory utilization for encryption and decryption processes for each algorithm, respectively, while Figure 9 and Figure 10 graphically interprets such values.

6. CONCLUSION

Data security and privacy have been always the main concern about cloud computing services. Cryptography is a main trend to achieve data security and privacy. Since the introduction of cloud computing technology, several cryptography solutions have been proposed for protecting outsourced data and user privacy and for insuring that data are not being leaked to unauthorized third party. Most of these solution schemes aim at achieving a tradeoff between security and functionality. This paper represents a comprehensive study of the major symmetric key block cipher algorithms, namely, DES, 3DES, Blowfish, Twofish, RC2, RC5, RC6, and AES. Each algorithm aims to introduce extra level of security and to satisfy performance requirements more than earlier proposed algorithms solution. This extra security level should balance between a robust algorithm structure and a reasonable complexity computation. This work provides a detailed review of the structure of the mentioned algorithms along with a performance evaluation comparison that highlights the strengths and limitations of each algorithm. From the tabular comparison, some key points can be concluded. First, the Twofish and Blowfish algorithms outperform the others followed by RC6 and AES, while 3DES and RC2 come at the tail of the list. Second, when it comes to memory usage, there is no significant difference between most of the algorithms except for 3DES which swept over almost three times of the memory needed for the other algorithms. Third, while AES is considered one of the most secure algorithms and no attacks have been reported against it, Twofish and Blowfish are considered the fastest schemes for both encryption and decryption.The main goal of any encryption algorithm is security. However, other performance metrics may determine the preference use of one algorithm over another. In this section,

Table 1. Comparision of Stmmetric Algothims Structure

Algorithm	**Structure metrics**		
	Block size	***Key size***	***Rounds***
DES	64 bits	56 bits (+8 parity bits)	16
3DES	64 bits	168, 112 or 56 bits	48 DES-equivalent rounds
Blowfish	64 bits	32–448 bits	16
Towfish	128 bits	128, 192 or 256 bits	16
RC2	64 bite	8–1024 bits; default 64 bits	16 of type MIXING, 2 of type MASHING
RC5	32, 64 or 128 bits	0 to 2040 bits	1-255
RC6	128 bits	128, 192, or 256 bits	20
AES	128 bits	128, 192 or 256 bits	10, 12 or 14 (depending on key size)

Table 2. Encryption Time

File Type	***File Size***	**Encryption Time in milliseconds**							
		DES	***3DES***	***Blowfish***	***Twofish***	***RC2***	***RC5***	***RC6***	***AES***
JPG	96 KB	569.32	626.58	511.82	511.39	634.66	582.12	516.61	552.67
TXT	116 KB	737.1	698.61	654.13	646.12	719.81	759.39	666.56	756.4
PDF	324 KB	1792.7	1930.89	1655.37	1646.31	1938.56	1818.06	1698.27	1735.09
PPT	1.29 MB	6658.48	8121.04	6513.87	6480.82	8131.14	6646.65	6539.34	6569.1
MP3	2.04 MB	10762.89	11589.72	10293.91	10279.16	11886.39	10561.62	10331.95	10612.97
MP4	2.13 MB	11189.7	12209.9	10610.74	10600.05	12503.65	11057.51	10797.49	10820.48
Average Time		5285.0316	5862.79	5039.97	5027.308	5969.035	5237.558	5091.703	5174.451
Throughput in MB/s		0.18869	0.17009	0.19786	0.198364	0.16706	0.19040	0.19585	0.19272

Table 3. Decryption Time

File Type	File Size	Decryption Time in milliseconds							
		DES	**3DES**	**Blowfish**	**Twofish**	**RC2**	**RC5**	**RC6**	**AES**
JPG	96 KB	546.52	601.45	488.45	485.12	607.75	493.01	487.47	492.23
TXT	116 KB	653.26	669.18	599.07	588.11	679.67	610.97	609.05	615.84
PDF	324 KB	1803.27	1938.56	1659.71	1617.94	2178.89	1682.85	1645.62	1660.3
PPT	1.29 MB	6740.09	6742.68	6519.72	6408.95	6807.16	6793.51	6589.45	6667.7
MP3	2.04 MB	10406.47	10830.28	10331.07	10330.41	10899.95	10493.83	10431.84	10449.53
MP4	2.13 MB	11211.71	11401.17	10741.26	10699.38	11405.26	10856.3	10867.01	10966.1
Average Time		5226.88	5363.88	5056.54	5021.65	5429.78	5155.07	5105.07	5141.95
Throughput in MB/s		0.19079	0.18591	0.19721	0.19858	0.18366	0.19344	0.19534	0.19394

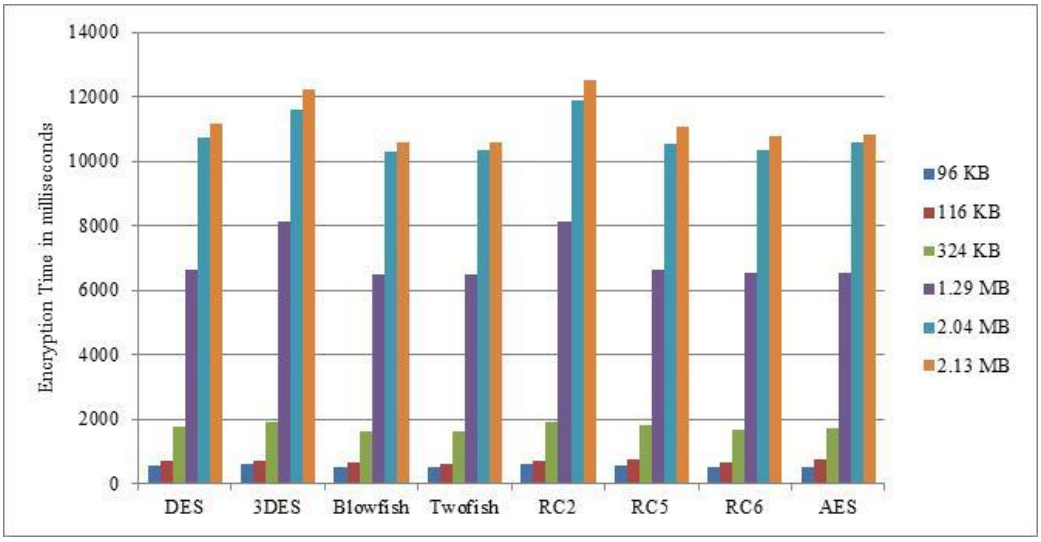

Figure 5: Encryption Time

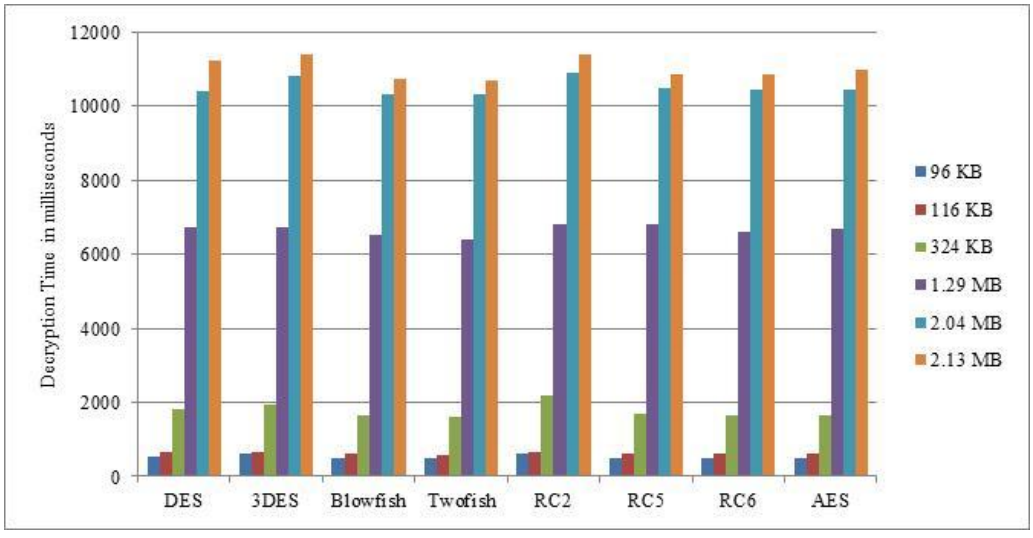

Figure 6: Decryption Time

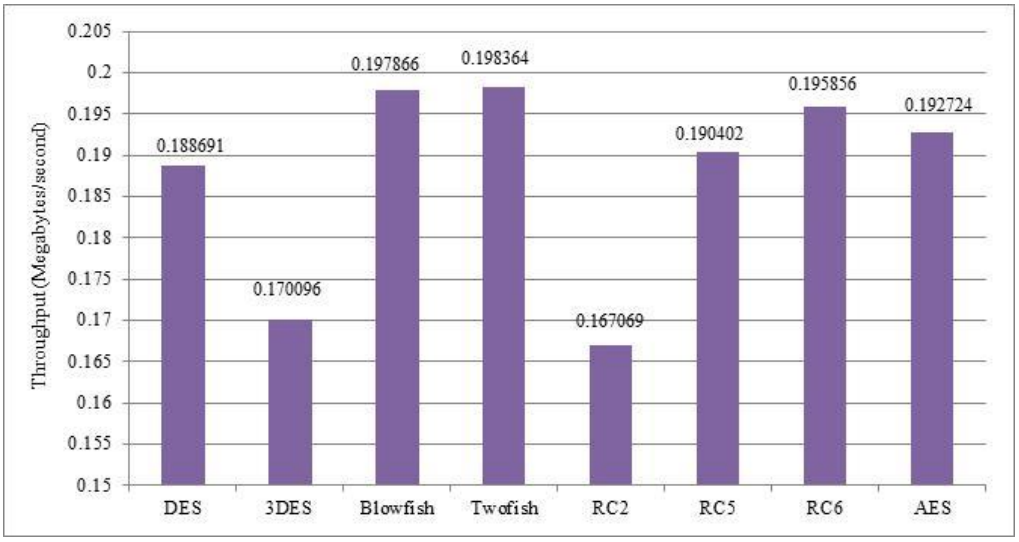

Figure 7: Throughput of encryption algorithm

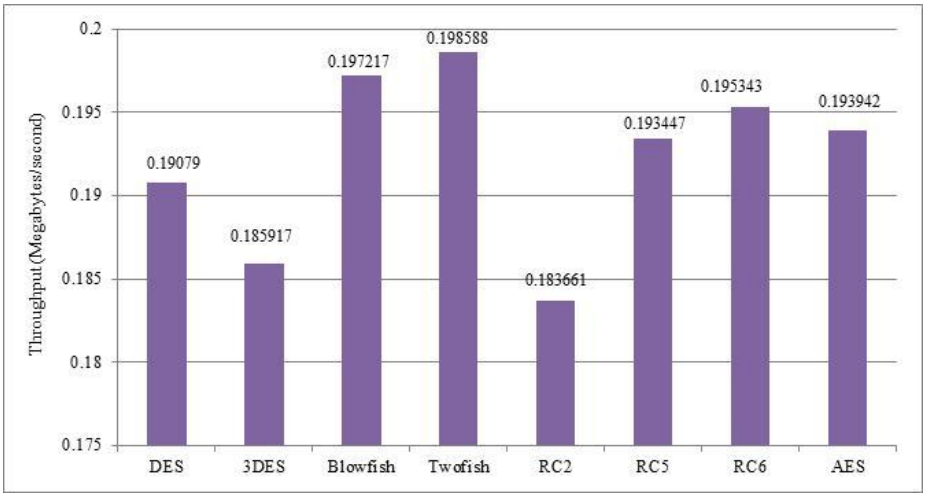

Figure 8: Throughput of decryption algorithms

Table 4. Memory Utilization for Encryption Algorithm

File Type	File Size	Memory Utilization for Encryption in Kilobytes							
		DES	3DES	Blowfish	Towfish	RC2	RC5	RC6	AES
JPG	96 KB	133.72	883.59	134.36	134.36	232.58	133.02	133.78	134.34
TXT	116 KB	224.62	883.59	225.7	225.7	324.84	225.59	224.73	225.68
PDF	324 KB	450.71	1343.13	452.85	452.85	551.13	340.13	450.91	340.29
PPT	1.29 MB	1456.36	3534.21	1463.37	1463.37	1957.78	1462.66	1457.07	1463.37
MP3	2.04 MB	2240.52	4417.75	2251.3	2251.3	2842.71	2250.21	2241.62	2251.3
MP4	2.13 MB	2342.27	5301.28	2353.6	2353.6	2944.61	2352.34	2343.54	2353.59

Table 5. Memory Utilization for Decryption Algorithm

File Type	File Size	Memory Utilization for Decryption in Kilobytes							
		DES	3DES	Blowfish	Towfish	RC2	RC5	RC6	AES
JPG	96 KB	123.53	883.59	124.13	126.16	142.38	122.79	122.32	124.13
TXT	116 KB	224.62	883.59	225.7	225.7	324.84	225.59	224.73	225.66
PDF	324 KB	448.15	1767.13	450.3	450.29	548.585	337.58	448.37	450.3
PPT	1.29 MB	1456.36	4417.75	1463.37	1463.37	1757.78	1462.66	1457.07	1463.37
MP3	2.04 MB	2189.59	7068.36	2200.21	2200.21	2891.77	2198.96	2190.68	2200.21
MP4	2.13 MB	2316.9	7068.36	2328.07	2328.05	2919.13	2326.85	2318.12	2328.07

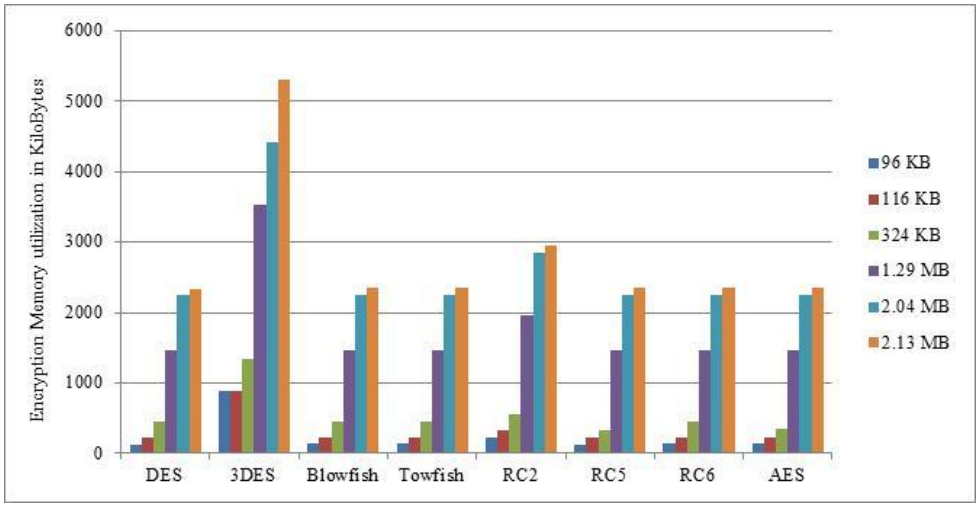

Figure 9: Memory Utilization for encryption algorithms

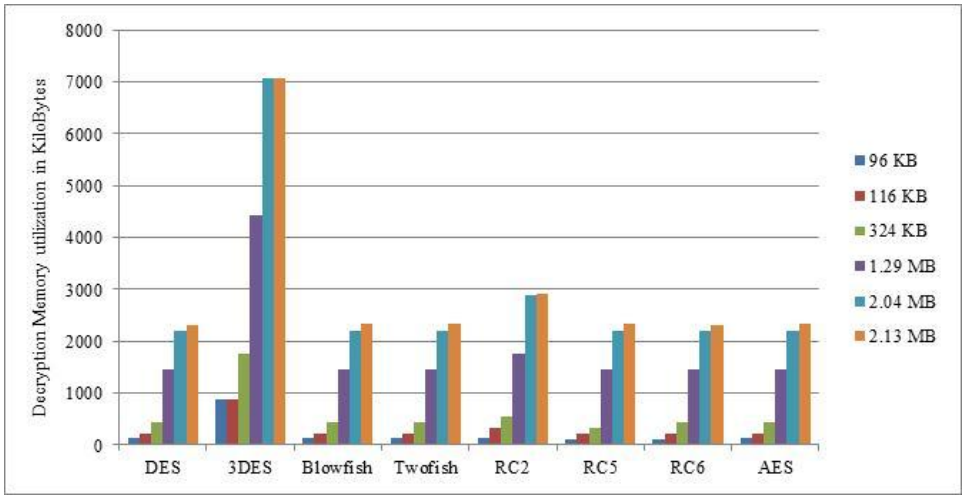

Figure 10: Memory Utilization for decryption algorithms

7. REFERENCES

[1] NIST SP 800-145, "A NIST definition of cloud computing", https://csrc.nist.gov/publications/detail/sp/800-145/final, September 2011.

[2] W. Stalling, "Cryptography and network security principles and practices," Fourth Edition, Prentice Hall, December 2006.

[3] H. Feistel, "Cryptography and computer privacy." Scientific American, vol. 228, no. 5, pp. 15-2, 1973.

[4] FIPS PUB 46-3, "Data encryption standared (DES)," National Bureau of Standards, U.S. Department of Commerce, January 1977.

[5] D. Coppersmith · D. B. Johnson · S. M. Matyas, "A proposed mode for triple-DES encryption," IBM Journal of Research and Development, April 1996.

[7] B. Schneier, "The Blowfish Encryption Algorithm," Dr. Dobb's Journal , v.19, n. 4, pp. 38-40, April 1994.

[8] B. Schneier, "Description of a New Variable-Length Key, 64-bit Block Cipher (Blowfish)," Fast Software Encryption: Second International Workshop, Leuven, Belgium, Proceedings, Springer-Verlag, pp.191-204, December 1994.

[9] L. Knudsen, V. Rijmen, R. Rivest, and M. Robshaw, "On the Design and Security of RC2", Fast Software Encryption, 5th International Workshop, FSE '98, Paris, France, March 23-25, 1998.

[10] R. Rivest, " The RC5 encryption algorithm", International Workshop on Fast Software Encryption, pp 86-96, 1994.

[11] B. Schneier, J. Kelsey, D. Whiting, D. Wagner, C. Hall, and N. Ferguson. "Twofish: A 128-bit block cipher." In AES Round 1 Technical Evaluation CD-1: Documentation. National Institute of Standards and Technology NIST, August 1998. See http://www. nist.gov/aes

[12] A. Mandal, C, Parakash and A. Tiwari, "Performance evaluation of cryptographic algorithms: DES and AES", IEEE Students' Conference on Electrical, Electronics and Computer Science, pp. 1-5, 2012.

[13] G. Singh and Supriya, "A study of encryption algorithms (RSA, DES, 3DES and AES) for information security," International Journal of Computer Applications, vol. 67, no. 19, pp. 33-38, April 2013.

[14] M. Faheem Mushtaq, S. Jamel, A. Disina, Z. Pindar, N. Shafinaz A. Shakir, M. Deris, " A Survey on the Cryptographic Encryption Algorithms", International Journal of Advanced Computer Science and Applications, vol. 8, no. 11, 2017.

[15] J. Daemen and V. Rijmen, "AES Proposal: Rijndael," AES algorithm submission, September 1999.

[16] M. Ebrahim, S. Khan, U. Khalid, "Symmetric algorthm survey: A comparative analysis". International Journal of Computer Applications, vol. 61, no. 20, pp. 12-19, January 2013.

[17] B. Schneier and D. Whiting, "A performance comparison of the five AES finalists." Proceedings of the Third AES Candidate Conference, pp. 123-135, April 2000.

[18] Z. Hercigonja and D. gimnazija, "Comparative analysis of cryptographic algorithms." International Journal of Digital Technology & Economy, vol.1, no. 2, 2016.

[19] D. Abdul Elminaam et al., "Performance Evaluation of Symmetric Encryption Algorithms", International Journal of Computer Science and Network Security, vol.8 no.12, December 2008.