Journal of Theoretical and Applied Information Technology
31st December 2018. Vol.96. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A PROACTIVE DATA SECURITY SCHEME OF FILES USING MINHASH TECHNIQUE

MEHDI EBADY MANAA [1], RASHA HUSSEIN JWDHA [2]

[1,2]Department of Information Networks, Faculty of Information Technology, University of Babylon, Iraq
Email: meh_man12@yahoo.com[1], rashahussein@itnet.uobabylon.edu.iq[2]

ABSTRACT

Data protection becomes an important issue for companies and organizations. Data security is one of the essential issues in the field of networks by synchronizing with the fast development of information technology, which is always seeking service providers to improve them in a way that serves the user and maintains the privacy of his data. It provides the goals of data security, which is confidentiality, availability, integrity. Cryptographic algorithms offer a healthy way for data confidentiality that used by many organizations using mainly two types of symmetric encryption and asymmetric encryption algorithms. The most important way to ensure the strength of the encryption algorithm is to generate the encryption key in robust ways. In this paper, the cryptosystem system is used the principle of the minhash technique to generate a group of keys in an efficient way. The block keys are generated using the k-shingle which is one of the main principles used in the minhash technique to convert the text file into a sequence of consecutive words. The length of shingles depends on the length of k value. The cryptosystem is used many hash functions to generate the cipher keys and then to encrypt text files using AES, DES, 3DES and Blowfish algorithms. The results show promised way to deny the hacker from unauthorized access. The AES and Blowfish algorithms have excellent results regarding encryption time, throughput, memory space, avalanche effect and entropy in term of security.

Keywords: *Algorithms of Cryptography, Minhash Technique, Data Encryption, AES Algorithm, The Blowfish Algorithm.*

1. INTRODUCTION

The cryptography is the art and science of converting essential information into a non-understandable form that can be not understood by the third part or attacker. The word Cryptography is derived from Greek origin, Crypto in the sense of secret and Graph in the sense of writing. The message in this science is converted from the readable form into the unreadable form and then sent to the recipient which he is only authenticated to decrypt the encrypted data or text using the agreed secret key shared by the sender and recipient [1][2].

The security criteria are Integrity, Availability, and Confidentiality (CIA) that can be conducted using the cryptography to protect the data for large organizations through access to the internet. It provides also authentication and non-repudiation for these organizations. In addition, these criteria provide an environment to use in different disciplines such mathematics, electrical engineering, computer science, and many applications such as computer password, ATM cards, message integrity techniques, digital signatures, secure computation, interactive proof, identity authentication, and electronic commerce[3].

It is necessary to pay attention to data security, which is called information security to concern the studying and methods of protecting data that is stored in computers devices and communications networks. The methods of information security are provided defense against attacks and unauthorized access to stored databases or those intended to transfer, alter or destroy the information in these databases[4].

Figure (1) shows the classification of encryption techniques and classification modern cipher techniques into two main categories:

Journal of Theoretical and Applied Information Technology
31st December 2018. Vol.96. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

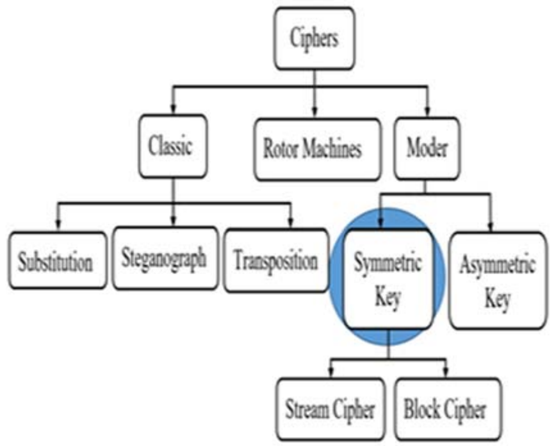

Figure 1: Classification of the Modern Ciphers Techniques[3].

The modern cipher categories can be classified into subcategories as below: -

I. The symmetric algorithms are used the shared secret key in the encryption and decryption process[5]. The main advantages of the symmetric algorithms are easy to use and fast to implement but one of the main drawbacks is to use these algorithms in large networks. Examples of algorithms that use the symmetric key are Data Encryption Standard(DES), Triple Data Encryption Standard(Triple DES/3DES), International Data Encryption Algorithm(IDEA), Advanced Encryption Standard(AES), Blowfish, and others [6]. The simple form of the symmetric encryption is conducted using encryption and decryption between two parties (Alice and Bob) using shared key between them as depicted in figure (2).

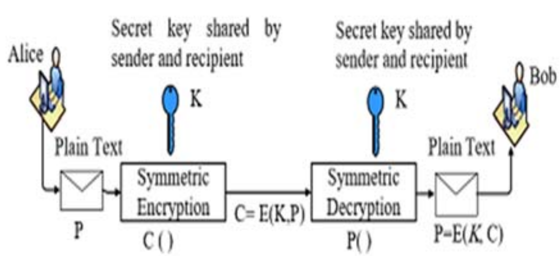

Figure 2: An approach of symmetric key cryptography

This form provides a secure communication between Bob and Alice against adversaries using one of the symmetric encryption algorithms to encrypted block/stream cipher.

II. The asymmetric key encryption (public key encryption) is a type of encryption where the user has a pair of encryption keys, public key, and the private key. The private key remains secret (special for the person that will decryption message). The public key shares between the parties and with everyone. The advantage of this type is that when the message is encrypted using the public key, it can only be decrypted by the corresponding private key. The main disadvantages of this type are slower than a symmetric algorithm. Examples of the asymmetric algorithm are Rivest-Shamir-Adleman(RSA), Digital Signature Algorithm(DSA), Diffie Hellman(DH) [7].

In this paper, we use block cipher based on a K-shingle with minhash technique to encrypt and decrypt the files that include sensitive information for the large organizations. The K-shingle is used to convert the text file into consecutive tokens depends on the length number of K. the minhash technique is used to generate many keys. These keys are manipulated as cipher keys for the symmetric algorithms such as AES, DES, 3DES, and Blowfish that used in this paper. The remaining of this paper is organized as follow. Section (2) presents the related work. The cipher method background in section (3). The key phase in section (4). The materials and results (proposed system) are illustrated in section (5). The results and discussion are shown in section (6).The comparison and limitation in section(7). Finally. Section (8) discussed the conclusion of the paper.

2. RELATED WORKS

The commonly related works for this work is illustrated in this part which is many of the research used symmetric algorithms to encrypt the sensitive data and then use the same key to decrypt it for original form.

The cost performance evaluation of cryptographic algorithms in term of encryption time, memory used, avalanche effect, entropy and number of bits for encoding optimality is proposed in [8]. They used many cryptographic algorithms DES, 3DES, AES, RSA, and blowfish. The key is generated by "KeyGenerator" object using packages in java security and java crypto. They experiment concluded the Blowfish algorithm is the appropriate algorithm regarding memory used and encryption time. The DES algorithm is the best in the bandwidth and AES is the best in cryptographic strength. The high level of avalanche effect is 70% in the results.

The comparison between AES (Rijndael), Triple DES, DES, RC6, and Blowfish algorithms are conducted using the different setting in [9]. They use many evaluations parameters data size, data type, encryption and decryption time, key size and power consumption. Simulation results show a comprehensive evaluation for each algorithm.

The proposed work in [10] is applied DES, 3DES, AES, and the Blowfish. The ECB (Electronic Codebook) and CFB (The Output Feedback) modes of operation are applied on different hardware platforms (Pentium-l 266 MHz machine and Pentium-4, 2.4 GHz machine). The results show in term of time that the Blowfish was the best.

In [11], a new key and S-box generation process were developed based on Self Synchronization Stream Cipher (SSS) algorithm. The key generation process for this algorithm was modified to be used with the blowfish algorithm. The experiment results for different size of data show low memory and less time but it faster than standard algorithm due to a total of 272 iterations to generate the subkeys compared to 521 iterations.

In addition, The proposed work in [12] evaluates AES and DES cipher algorithms. The evaluation parameters are conducted using processing time, CPU usage and throughput using two platforms in laptop core I5, 2.5 GH CPU on Windows 7 and Mac platform for different data size. The simulation results are evaluated for different sizes of files.

The modified Blowfish algorithm is designed and implemented [13] for networking and communication application for enhanced network security and defense applications. They use single Blowfish round instead of many rounds. They use Xilinx ISE platform for evaluating the work based on VHDL language.

Also, the comparisons between the two algorithms blowfish and Skipjack are designed and implemented for encrypting input files of varying contents and sizes. The results show Blowfish is the best performing algorithm for implementation[14].

The AES algorithm is implemented for five different execution platforms by F. García in [15]. The main objective of this work is to provide a limitation on the model for the configuration parameters of AES implementation.

In another study, various encryption algorithms (AES, Blowfish, Twofish, DES, RSA, and Diffie-Hellman) based on a different parameter are compared to choose the best data encryption algorithm by Gaurav and Aparna[16]. Simulation results are given to show the effectiveness of each algorithm with the encryption and decryption time, encryption speed and throughput.

3. THE BLOCK CIPHER ALGORITHMS

The most block cipher algorithms that used in this work are presented in the following sections. The work in implemented and analysis using Java. In spite of the Java compiler is generated Java Virtual Machine (JVM) that needs an interpreter to run it inside the machine code, the cipher algorithms show good results in term of many performance evaluation parameters and minhash technique.

3.1. Advanced Encryption Standard (AES)

AES was known as Rijndael, is one of symmetric key block cipher developed by Joaen Daemen, Vincent Rijmen . The block size is 128 bits and it has three key sizes 128, 192 and 256 bits. The AES rounds are 10 when key size 128, 12 and 14 rounds with key size 192 and 256 respectively. AES uses key expansion to encrypt/ decrypt data where key expansion comes before the encryption process and before the decryption process. It comes the advance of DES and 3DES [17].

The algorithm works by numbers of stages and started its work by Add Round Key stage and follow by 9 rounds of four stages. The encryption phase of AES algorithms work by four stages are as follows

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

3.1.1. Substitute bytes is illustrated by the algorithm 1.

Algorithm 1: Pseudocode for Sub Bytes Transformation
Input: S (State is array of plaint text)
Output: S1(Out State)

```
Begin
    SubBytes(S)
    {
        for(r=0 to 3)
            for (c =0 to 3)
                S1=subbyte(S_{r,c})
    }
    Subbyte(byte)
    {
        a ← byte^-1    //Multiplicative inverse in GF(2^8) with inverse of 00 to be 00
        ByteToMatrix (a,b)
        for (i = 0 to 7)
        {
            c_i ← b_i ⊕ b_{(i+4) mod 8} ⊕ b_{(i+5) mod 8} ⊕ b_{(i+6) mod 8} ⊕ b_{(i+7) mod 8}
            d_i ← c_i ⊕ ByteToMatrix (0x63)
        } end for
        MatrixToByte(d,d)→ S1
        byte ←d
    } End
```

3.1.2. Shift Rows is illustrated by the algorithm 2

Algorithm 2: Pseudocode for Shift Rows transformation
Input: S1 (State that out from SubByte)
Output: S2(out State)

```
Begin
    ShiftRows(S1)
    {
        for (j =1 to 3)
            S2=Shiftrow(S_j, j)    //sj is the rth row
    }
    //InvShift Rows
    Shiftrow(r,n)    // n is number of bytes to be shifted
    {
        CopyRow(r,tem)  //tem is temporary row
        for ( col=0 to 3)
            r_{(col-n) mod 4} ← tem_col
    }End
```

3.1.3. Mix columns are illustrated by the algorithm3.

Algorithm 3: Pseudocode for Mix Columns Transformation
Input: S2 (output state of Shift Rows)
Output: S3 (output state)

```
Begin
    MixCol(S2)
    {
        for (col=0 to 3)
            S3=mixcol(S2_col)
    }
    mixcol ( c )
    {
        Copycol(c,tem) //tem is a temporary column
        c_0 ← (0x02) • tem_0 ⊕ (0x03 • tem1) ⊕ tem2 ⊕ tem3
        c_1 ← tem_0 ⊕ (0x02) • tem_1 ⊕ (0x03) ⊕ tem_2 ⊕ tem_3
        c_2 ← tem_0 ⊕ tem_1 ⊕ (0x02) • tem_2 ⊕ (0x03) • tem_3
        c_3 ← (0x03 • tem_0) ⊕ tem_1 ⊕ tem_2 ⊕ (0x02) • tem_3
    } End
```

3.1.4. Add round key is illustrated by the algorithm 4.

Algorithm 4: Pseudocode for AddRoundkey Transformation
Input: S3 (State) , W (word is sub keys)
Output: S4 (Cipher state)

```
Begin
    AddRoundKey(S3) {
        for( c=0 to 3)
            S4 = S3_c ← S3_c ⊕ w_{round+4c}
    }
    End
```

3.2. Data Encryption Standard

The DES was published by National Institute of Standards and Technology (NIST) as first encryption standard. It was found by IBM based on their Lucifer cipher which considers as block cipher algorithm with 64 bits key length as standard. The NSA puts a restriction to use 56 bits as key length for block cipher encryption with 64 bits and discard 8 bits from 64-bit as the length of the key. The DES is more flexible because it works with different modes such as Output FeedBack (OFB), Cipher Block Chaining (CBC) ...etc. [18].

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

The main drawback of DES is used 56 bits for key length and it was broken by supercomputer for a total of 22 hours by DES cracker in 1998 [8]. For this reason, it was modified to a 3DES algorithm, which was used the three-time key to increasing the strength of the algorithm but it becomes slow in time [19]. DES work following algorithm 5.

Algorithm 5: Pseudocode for DES algorithm

Input: p[64](plain text block size is 64 bit), RK (Round key is 16 rounds and 48 bit), Fun()(round function)

Output: oB (output cipher block)

```
Begin {
    Cipher( p[64],RK[16,48], c[64])  // c is cipher text block is 64 bit
    {
        Per (64,64,p,iB,IP Table) //iB is inBlock , IP Table is InitialPermutation Table
        Spilt(64,32,iB,lB,rB)  //lB is leftBlock and rB is rightBlock
        for(rod = 1 to 16)  //rod is round
        {
            Mix (lB,rB,RK[rod])
            If(rod != 16)swp (lB,rB)
        }
        Cmb (32,64,lB,rB,oB)
        Per (64,64,oB,c,FP Table) // oB is outBlock and FP table is FinalPermutation Table
    }
    Mix (lB[48],rB [48],RK[48])
    {
        Copy(32,rB,T1)
        Fun (T1,RK,T2)
        XOR(32,lB,T2,T3)
        Copy( 32,T3,rB)
    }
    Swp (lB[32] ,rB [32])
    {
        Copy(32,lB,T)
        Copy(32,rB,lB)
        Copy(32, T,rB)
    }
    Fun (iB[32],RK[48],oB [32])
    {
        per (32,48,iB,T1,EP Table) // EP table is Expansion Permutation Table
        XOR(48,T1,RK,T2)
        subs (T2,T3,Subs Tables)  // Subs Table is Substitute Tables
        per (32,32,T3,oB,SP Table) // SP Table is Straight Permutation Table
    }
    subs (inB[32], oB[48], Subs Tables[8,4,16])
    {
        For(i= 1 to 8)
        {
            row ← 2 ×iB[ i ×6+1]+iB[i ×6+6]
            col ← 8 ×iB[ i ×6+2]+4 ×iB[i ×6+3]
            2 ×iB[ i ×6+4]+iB[i ×6+5]
            value=Subs Tables[i][row][col]
            oB[[i ×4+1]←value⁄8; value ←value mod 8
            oB[[i ×4+2]←value⁄4; value ←value mod 4
            oB[[i ×4+3]←value⁄2; value ←value mod 2
            oB[[i ×4+4]←value
        }//end for
    }// end substitute
}// End Algorithm
```

3.3. Triple Data Encryption Standard (3DES/Triple DES)

The enhanced version of DES without designing a whole new cryptosystem. It is the symmetric key block cipher algorithm (3DES). It was first published in 1998. The name of the 3DES comes from applying three times DES cipher. The main problem of 3DES is slower in the performance than DES but is much more secure. The main advantage was to increase the key size to provide more secure for the ciphers data with 112- and 168-bit length key for block cipher 64 bits [20].

3.4. Blowfish Algorithm

Blowfish is symmetric key block cipher algorithm with block size 64 bits and variable key cipher length from 32 to 448 bits to protect the encrypted data. The structure of this algorithm is the fiestal network. The algorithm was first introduced in 1993 by Bruce Schneier and has not been cracked yet. It can be optimized in hardware applications due to its compactness. The details of this algorithm can be found in[21]and Its work is illustrated by the following algorithm 6.

Algorithm 6: Blowfish Decryption Algorithm
Input: C (cipher text is 64 bit); P1, P2, ..., P18(P-array is 18 sub keys) , fun( )(Round function)
Output: T (64 bits of clear text)
Begin 1. Divide C into two 32bit halves: C_L, C_R 2. for j = 1 to 16 3. $C_L = C_L \oplus P_{(19-j)}$ 4. $C_R = \mathrm{fun}\,[C_L] \oplus C_R$ 5. Swap C_L and C_R 6. Next j 7. Undo the last swap, Swap C_L and C_R 8. $C_R = C_R \oplus P_2$ 9. $C_L = C_L \oplus P_1$ 10. T = Recombine C_L and C_R 11. Calculate Function fun 12. C_L is divided into 4 eight-bit quarters: A1, A2, A3, and A4 13. fun [C_L]= ((S1[A1]+ S2 [A2] mod 2^{32}) $\oplus$ S3[A3]) + S4[A4] mod 2^{32} 14. End

4. KEY GENERATION METHODS

The main method to generate the keys in this work is described by the following subsections

4.1 K-shingle(N-grams)

The term of K-shingle is used heavily in document similarity, in this technique, documents are split into a set of tokens depending on the length of k. for example, if the document has the string "The weather is nice and the sky is blue". If we choose k=3, the number of the generated tokens is equal $(n-k+1)$ where n is the total number of documents words k is the shingle length. The generated tokens are {“The weather is, weather is nice, is nice and, nice and the, and the sky, the sky is, sky is blue”}. In this paper, we split the text files into shingles based on the number length of k. then we applied the hash functions (MD5, SHA-512, and SHA-256) for each shingle. The minhash technique is applied for the hashed tokens

4.2 Minhash Function

The principle of minhash technique is applied in this work. The general form of the minhash function is given in equation (1)[22].

$$h(x) = ax + b \bmod c \quad \dots.(1)$$

where a and b are two random values, x is the hash function value for the tokens and **c** is a prime number that is great than the maximum number of x [23]. For example, let we use 4 randomly generated minhash functions as shown in equations (2 to 5) respectively for supposing have shingles (C, D, F, E) and by applying four equations the results show in a table (1).

$$(12x+8)\ mod\ 17 \quad \dots(2)$$
$$(14x+3) mod 17 \quad \dots(3)$$
$$(11x+5)\ mod\ 17 \quad \dots(4)$$
$$(16x+7)\ mod\ 17 \quad \dots(5)$$

Suppose the real hash values for the first shingle is C and equal
C=10101001101011100110110111010110110000111000101001100110011110010111101111110101011001111010010100011000101000001010011010 11010

D=101000010111110100000000001110010010000101100100110000101000101110101001101010001001011010011001000101110011101001001000001111

F=111101010110101111010000100111100001100111110001010011000001100111011101011101011001001010110101000000101010101110010000101010 10

E=100110111101000100011100100011010010100011010010101100001011110010011001010100011111100001101111010011011100111100100110011 10

The hash functions in above are used to calculate the value of each hash token for the file text as depicted in a table (1) below.

Table 1: The Proposed Minhash work

# The hashed shingles	Minhash_2	Minhash_3	Minhash_4	Minhash_5
C	5	0	16	6
D	14	14	10	5
F	6	13	8	16
E	15	8	15	3

The outputs values are hashed again to (128 bits) using one hash function as MD5 or other to be the keys for cipher block algorithms. In this work, we used the same steps to generate the keys using one hash functions and applied on data.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

5. THE PROPOSED SYSTEM

In this work, we have implemented and compared four block cipher algorithms DES, 3DES, AES, and Blowfish. We have implemented algorithms in Java and using files in different sizes. The keys generation are applied based on K-shingles and minhash. The evaluation metrics are applied based on the encryption time, memory usage, entropy, throughput, and avalanche effect to evaluate the performance of these algorithms. The system results are explained in detail in the following subsections. The proposed system is illustrated in figure (3)

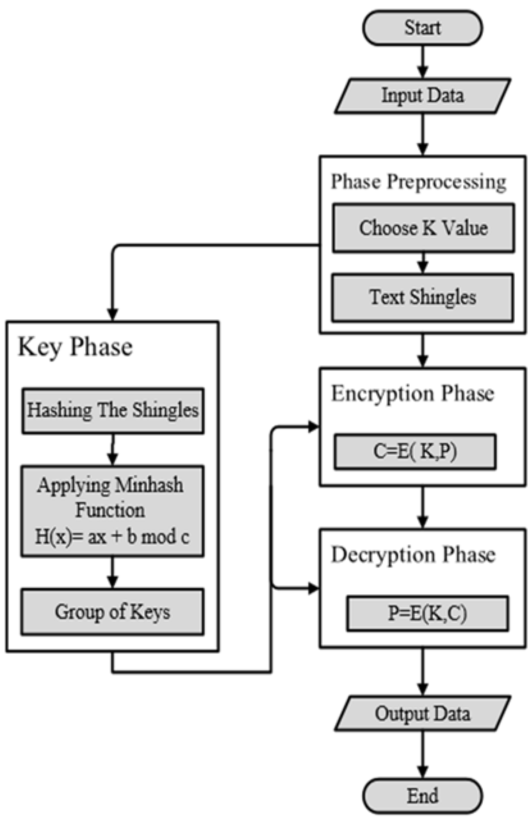

Figure 3: The Encryption/Decryption Files Proposed System.

5.1. The Key Generation Phase

At this phase, the K-shingle, hash function and minhash technique are used. K-shingle is the process of split the input of the data into substring according to the specified length of K. The punctuation marks and spaces are removed before applying the shingling process. The hash function MD5 is applied for each generated shingle.

The minhash function is implemented for each hashed shingle. In this work, we choose 10 hash functions to test the results which are the text files such as .pdf, .docx, and .txt in the different size. The values of a and b are chosen randomly. The value of the x is represented the minimum value after applied the 10 hash functions in randomly setting for a and b values. The output of this stage is many keys for each shingle that used later for cipher process.

5.2. The Encryption/Decryption Phase

The generated keys from previous phases are applied for each block size text (n-k+1). For example, the key 1 is applied to block cipher 1 and key 2 is applied for block cipher 2 and so on.

In the decryption, the same keys are used for the encrypted text to convert it to its original form.

The main pseudo code steps for the proposed system is illustrated in the algorithm (7)

Algorithm 7: The Proposed System Algorithm
Input: D (Input Data), $Ki = \{K1, \ldots, Kn\}$: $k_i \subset g_i\ (ax_i + b\ mod\ c)$ (Group of Keys
Output: E_file(Encryption file)

Begin
1. Apply k-shingle on D
2. Set K value for K-shingle
3. Calculate Function k-shingle (k ,D){
4. Read (D)
5. Word [ ] =Preprocessing(D)
6. shingles = split Word [ ] into shingles based on (n-k +1) length
7. return shingles
8. End
9. Apply Hash function
10. Calculate function hash (shingles){
11. foreach shingles in k-shingle (k, D)
12. hash value = Hashing shingles (MD5(shingles))
13. x= Convert hash value to decimal
14. return x
15. } // end hash function
16. Apply Minhash equation (Keys (K_i)) $(ax_i + b)\ mod\ c$
17. Calculate Function Minhash {
18. Repeat
19. foreach shingles in K-shingle (k, D)
20. $for\ i \leftarrow 1\ to\ 10\ do$
21. $Random\ rnd = new\ Random();$ $a\ and\ b = rnd.nextint(100);$
22. $If\ (c > x\ \&\&\ c > a\ \&\&\ c > b\ \&\&\ c\ is\ checked\ prime)\ then$ //a,b,c is parameters of minhash equation(1)
23. $h(x_i) = (ax_i + b\ mod\ c)$
24. else
25. return to step20

Journal of Theoretical and Applied Information Technology
31st December 2018. Vol.96. No 24

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

26. end if
27. end for
28. $return\ min\ (h(x_i))$
29. $\}\ end\ foreach$
30. Apply hash function again
31. k_i=MD5($min\ (h(x_i))$)
32. $return\ ki$
33. Until (End Minhash calculate of one shingle)
34. } End Minhash Function
35. Choose Cryptography algorithm (DES,3DE, AES, Blowfish)
36. Encryption Phase
37. Pi= Shingling based on the number on K that choose in 2
38. Generate Cipher Text E_file = E (Ki, Pi)
39. Evaluation Cipher algorithm choose using Time , Throughput, Memory ,Avalanche effect and Entropy {
40. Calculate Time((Start time – End time)
41. Calculate Throughput
42. Throughput=(D size in byte)/(Encryption Time (ms))
43. Calculate Memory as
44. Total Memory = (Total memory of instance – Free memory of instance /1024)
45. Calculate Avalanche effect
46. Avalanche effect=(no.of flipped bits in E_file)/(no.of bits in E_file)
47. Calculate Entropy for all Shingles in E_file
48. $H(x_i) := -\sum_{j=0}^{r-1} P_j log\ P_j$ // r is a number of Shingles based ((n-k)+1)
49. } End evaluation
50. End Algorithm

6. Result and Discussion

The main criteria to evaluate the algorithms (DES, Triple DES, AES, and Blowfish), that are implemented in this work, are encryption time, memory usage, entropy, throughput and avalanche effect [8][24]. These results are all used K=5.

6.1. Encryption Time

Encryption time is the amount of time that the algorithm is needed to convert data from plaintext to ciphertext depending on the key size and data block size. The less time the algorithm takes, the better the algorithm is used to embed encryption it in other applications such as e-commerce, banking, and online transaction processing applications. In this work, we measure the time in milliseconds for different size of input files in bytes

Table 2: The Encryption Time at K=5 in K-shingle

Input Size (Bytes)	Times in Millisecond			
	DES	Triple DES	AES	Blowfish
114	85	123	2	1.23
1,164	243	636	11.43	4.99
2,492	321	756	21.8	8.21
12,571	1567	2341	144	14.14
24,490	703	3696	564	23.7
1,278,548	69231	108731	18003	587
2,240,213	85503	174378	32897	1124
3,600,365	98102	341260	53353	1417

It is noticed from the figure (4) that the Blowfish and AES take less time for encryption, while DES and 3DES need a long time.

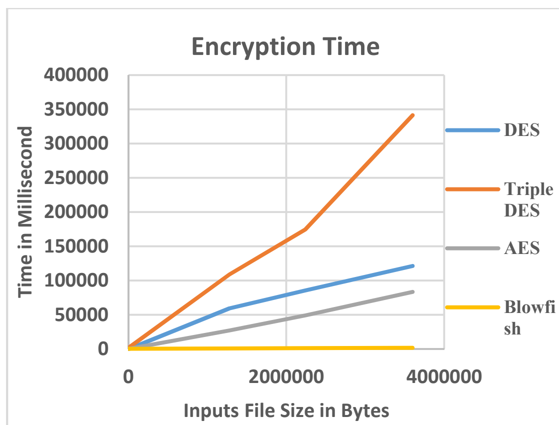

Figure4: Encryption Time for DES, Triple DES, AES, and Blowfish.

6.2. Throughput

The Throughput is calculated by using the equation (6)

$$Throuphput = \frac{Plaintext\ file\ size\ in\ byte}{Encryption\ Time\ (ms)} \quad .... (6)$$

Table (3) shows the throughput values for different for the four-block cipher symmetric algorithms.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table3: Throughput Values for DES, 3DES, AES, and Blowfish

Input Size (Bytes)	Throughput (b/Ms)			
	DES	Triple DES	AES	Blowfish
114	1.341176471	0.926829268	57	92.68292683
1,164	4.790123457	1.830188679	101.8372703	233.2665331
2,492	7.763239875	3.296296296	114.3119266	303.5322777
12,571	8.022335673	5.369927381	87.29861111	889.0381895
24,490	34.83641536	6.626082251	43.42198582	1033.333333
1,278,548	18.467854	11.75881763	71.01860801	2178.105622
2,240,213	26.20040233	12.84687862	68.09779007	1993.072064
3,600,365	36.70022018	10.55021098	67.48195978	2540.836274

It is noticed in figure (5), the two algorithms Blowfish and AES satisfy the best throughput values.

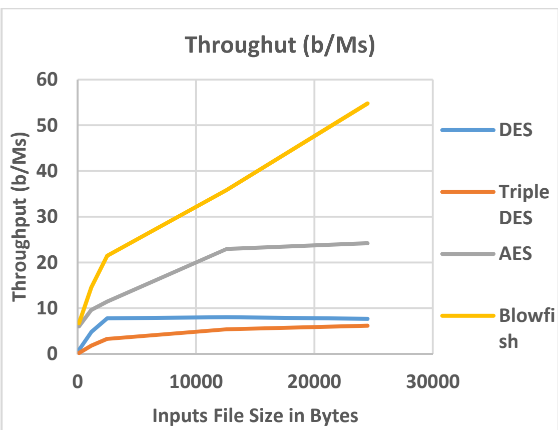

Figure 5: Throughput (b/Ms) using Encryption Algorithms.

6.3. Memory used

The memory usage is the space that is reserved for implementation the algorithms which depend on the number of operations in the algorithm, key size, initialization vectors and type of mode operations. It is noticed from the memory usage results that the Blowfish algorithm is lower consumption of memory, as shown in the table (4).

Table 4: Memory usage for Cryptography algorithms

Input Size (Bytes)	Memory Usage (KB)			
	DES	Triple DES	AES	Blowfish
114	6101	7117	456	279
1,164	6899	8297	5162	4000
2,492	11430	13010	9206	7240
12,571	15631	20498	11192	9406
24,490	54988	79552	34045	28577
1,278,548	464104	805563	254758	161661
2,240,213	861668	1275626	779425	618351
3,600,365	929829	1392101	566144	692257

It is noticed for the figure (6) that the Blowfish has the least memory consumption, while the DES algorithm has the highest memory consumption.

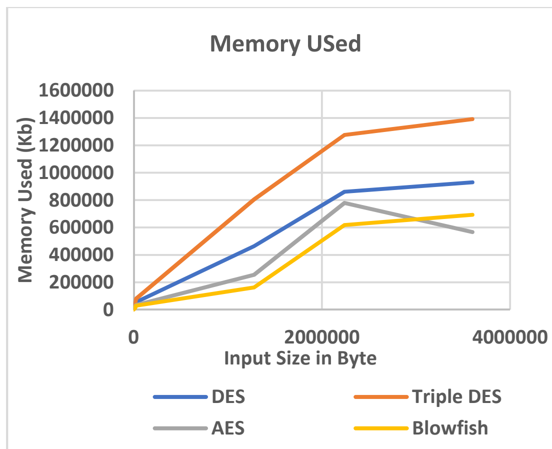

Figure 6: Memory Usage in Kb for four Encryption Algorithms.

6.4. *Avalanche effect*

Equation (7) is used to calculate the avalanche effect by using the Hamming distance.

$$Avalanche\ effect = \frac{no.of\ flipped\ bits\ in\ the\ ciphered\ text}{no.of\ bits\ in\ ciphered\ text} \quad \dots (7)$$

It can be calculated using the equation (8).

$$Avalanche\ effect = \frac{no.of\ flipped\ bits\ in\ ciphered\ text}{no.of\ bits\ in\ ciphered\ text} \quad \dots (8)$$

It is noted from the table (5) using the equation (8) that the AES has the highest values and

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

satisfy the best results than DES which has the least value.

Table 5: Average of Avalanche effect in the plain text

No. of fillipe d bits	The Proposed Method (%)				
	DES	DES	DES	DES	DES
1	17.1875	17.1875	17.1875	17.1875	17.1875
2	17.708	17.708	17.708	17.708	17.708
4	0	0	0	0	0
19	0	0	0	0	0
35	33.680	33.680	33.680	33.680	33.680
45	26.988	26.988	26.988	26.988	26.988
Average	15.927	15.927	15.927	15.927	15.927

It is cleared that the proposed method satisfy good results when we compare with results in [25][26] and)[27] in terms of these cipher algorithms.

6.5. *Entropy Value*

It is calculated using the Shanon law in equation (9). It is a measure of randomness which indicates that the power of the cipher algorithms against attack.

$$H(x) = -\sum_{i=0}^{n-1} p(xi)\log\big(p(xi)\big) \ldots \quad (9)$$

H(x) is entropy of x.

It is cleared from the table (6) that the AES and Blowfish have the highest values in randomizing.

TABLE 6: AVERAGE OF ENTROPY VALUES FOR CIPHER ALGORITHMS

Input Size(Byte)	Entropy Value			
	DES		DES	
1,164	7.45121	7.45121	7.45121	7.45121
2,492	8.50779	8.50779	8.50779	8.50779
12,571	10.86936	10.87574	10.87574	10.87574
114	3.96981	4.08746	4.08746	4.08746
24,490	10.54109	11.54109	11.54109	11.54109
1,278,548	12.42301448	17.20865431	12.42337858	12.42337858
2,240,213	13.51286389	18.19757503	13.51286389	13.51286389
3,600,365	13.40274562	13.40274562	13.40274562	13.40274562

The Blowfish takes the smallest time in the encryption and the Triple DES takes more time in the encryption. If the speed is important in the application, the Blowfish is the best option and has the highest value in throughput. If the memory required for implementation is small, the blowfish is the best option and the 3DES is not preferable. Avalanche effect for the AES is high. Thus, it can be used for the applications where the confidentiality and integrity are of highest priority. Also, Blowfish and 3DES have high values of the avalanche effect. The entropy values for all algorithms have an efficient random value that is robust against guessing attacks.

7. THE COMPARISON AND LIMITATION

The comparison and limitation of the proposed cryptosystem show a promising tool for the cryptosystem when it is compared with related works in the table (7).

The limitation occurs when the data size is grown. Map-reduce is a more promising programming model when the data is huge.

8. CONCLUSION

The power of cipher algorithms is mainly depending on the strength of the generated key or key length for cipher algorithms. This paper comes to propose a robust key generation based on the principles of K-shingle and minhash technique. The AES, DES, 3DES and Blowfish algorithms are implemented in this work to cipher various files based on the minhash technique. We conclude many results in term of many performance evaluations such as encryption time, throughput, memory used, avalanche effect, and entropy. It is concluded that the AES and Blowfish superior the other used algorithms. In the other results, we conclude that the DES and 3DES satisfy fewer results in term of encryption time, throughput and memory consumption where they require a lot of time to execute. In addition, the obtained results show the Blowfish is more suitable for the applications that need speed and AES is better for the applications that need confidentiality and integrity with the highest priority. The strength of the cipher algorithms in this work shows that entropy has the equal value which indicates the randomness strength in these cipher algorithms.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

REFERENCES

[1] B. Schneier, *Applied cryptography: Protocols, algorithm, and source code in C*. John Wiley & Sons, Inc., New York, NY, 1996.

[2] M. Ahmed, T. M. S. Sazzad, and M. E. Mollah, "Cryptography and State-of-the-art Techniques," *IJCSI International Journal of Computer Science Issues*, vol. 9, no. 2–3, 2012.

[3] K. Goyal and S. Kinger, "Hybrid Approach Using Encryption Algorithms For Data Storage," *International Journal of Scientific & Engineering Research*, vol. 4, no. 7, pp. 2063–2067, 2013.

[4] M. E.Whitman and H. J. Mattord, *Principles of Information Security Fourth Edition*, Fourth Edi.

[5] A. Hussein Zyara, "Suggested Method for Encryption and Hiding Image using LCG and LSB," *Journal of University of Babylon*, vol. 26, no. 2, pp. 66–77, 2017.

[6] H. A. Lafta and R. Abdul-Wahid M. Ali, "Symmetric Cryptosystem Based on Petri Net," *Journal of University of Babylon*, vol. 26, no. 1, pp. 1–8, 2017.

[7] P. Mahajan and A. Sachdeva, "A Study of Encryption Algorithms AES, DES and RSA for Security," *Global Journal of Computer Science and Technology Network, Web & Security*, vol. 13, no. 15, 2013.

[8] P. Patil, P. Narayankar, D. G. Narayan, and S. M. Meena, "A comprehensive evaluation of cryptographic algorithms: DES, 3DES, AES, RSA and Blowfish," *Elsevier B. V. Procedia Computer Science.*, vol. 78, pp. 617–624, 2016.

[9] D. Salama Abdul. Elminaam, H. Mohamed Abdul Kader, and M. Mohamed Hadhoud, "Performance evaluation of symmetric encryption algorithms," *IJCSNS International Journal of Computer Science and Network Security*, vol. 8, no. 12, pp. 280–286, 2008.

[10] A. Nadeem and M. Y. Javed, "A Performance Comparison of Data Encryption Algorithms," in *In Information and communication technologies, 2005. ICICT 2005. First international conference on (pp. 84-89). IEEE.*

[11] T. S. Atia, "Development of A New Algorithm for Key and S-Box Generation in Blowfish Algorithm," Journal of Engineering Science and Technology, Vol. 9, No. 4, pp. 432–442, 2014.

[12] S. D. Rihan, A. Khalid, and S. Eldin F. Osman, "A Performance Comparison of Encryption Algorithms AES and DES," *International Journal of Engineering Research & Technology (IJERT)*, vol. 4, no. 12, pp. 151–154, 2015.

[13] S. Manku and K. Vasanth, "Blowfish encryption algorithm for information security," *ARPN Journal of Engineering and Applied Sciences*, vol. 10, no. 10, pp. 4717–4719, 2015.

[14] A. Ahmad Milad, H. Zaiton Muda, Z. A. Bin Muhamad Noh, and M. Almahdi Algaet, "Comparative Study of Performance in Cryptography Algorithms (Blowfish and Skipjack)," *Journal of Computer Science*, vol. 8, no. 7, p. 91, 2012.

[15] D. F. García, "Performance Evaluation of Advanced Encryption Standard Algorithm," in *Mathematics and Computers in Sciences and in Industry (MCSI), 2015 Second International Conference on. IEEE,* 2015, pp. 247–252.

[16] G. Yadav and A. Majare, "A Comparative Study of Performance Analysis of Various Encryption Algorithms," in *International Conference On Emanations in Modern Technology and Engineering*, 2017, vol. 5, no. 3, pp. 70–73.

[17] B. A.Forouzan, *Cryptography and Network Security*, 1st Editio. McGraw-Hill Education, 2007.

[18] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 6th Editio. NJ, Englewood Cliffs:Prentice-Hall, 2006.

[19] C. Paar and J. Pelzl, *Understanding cryptography: a textbook for students and practitioners*. 2009.

[20] N. Aleisa, "A Comparison of the 3DES and AES Encryption Standards," *International Journal of Security and Its Applications*, vol. 9, no. 7, pp. 241–246, 2015.

[21] T. Nie and T. Zhang, "A study of DES and blowfish encryption algorithm," in *Tencon 2009-2009 IEEE Region 10 Conference*, 2009, pp. 1–4.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[22] J. Leskovec, A. Rajaraman, and J. D. Ullman, "Mining of Massive Datasets," 2014.

[23] S. Reddy Nagireddy, "Scalable Techniques for Similarity Search," 2015.

[24] C. Prakash Oewangan, S. Agrawal, A. Kumar Mandae, and A. Tiwari, "Study of avalanche effect in AES using binary codes," in *Advanced Communication Control and Computing Technologies (ICACCCT), 2012 IEEE International Conference on*, 2012, pp. 183–187.

[25] C. Echeverri, "Visualization of the Avalanche Effect in CT2," University of Mannheim, 2017.

[26] H. Agrawal and M. Sharma, "Implementation and analysis of various symmetric cryptosystems," *Indian Journal of science and Technology*, vol. 3, no. 12, pp. 1173–1176, 2010.

[27] A. Kumar and N. Tiwari, "Effective implementation and avalanche effect of AES," *International Journal of Security, Privacy and Trust Management (IJSPTM)*, vol. 1, no. 3/4, pp. 31–35, 2012.

Table 7: The comparison of the Proposed Cryptosystem with the Others works

Paper No	File Size	Method	Evaluation Parameters				
			Time	Throughput	Memory/CPU usage	Avalanche effect	Entropy
[8]	25KB	DES	≈485ms	NA[1]	18.2	≈60%	2.9477
		3DES	≈480ms	NA	20.7	≈50%	2.9477
		AES	≈510 ms	NA	14.7	≈70%	3.84024
		BF[2]	≈40ms	NA	9.38	≈37%	3.93891
	1Mb	DES	≈600ms	NA	NA	NA	NA
		3DES	≈600ms	NA	NA	NA	NA
		AES	≈590ms	NA	NA	NA	NA
		BF	≈500ms	NA	NA	NA	NA
	2Mb	DES	≈1010ms	NA	NA	NA	NA
		3DES	≈1360ms	NA	NA	NA	NA
		AES	≈590ms	NA	NA	NA	NA
		BF	≈505ms	NA	NA	NA	NA
	3Mb	DES	≈1400ms	NA	NA	NA	NA
		3DES	≈1800ms	NA	NA	NA	NA
		AES	≈650ms	NA	NA	NA	NA
		BF	≈1000ms	NA	NA	NA	NA
[10]	20527	DES	24 sec	NA	NA	NA	NA
		3DES	72sec	NA	NA	NA	NA
		AES	39sec	NA	NA	NA	NA
		BF	19sec	NA	NA	NA	NA
[11]	1KB	BF	7sec	NA	NA	NA	NA
	2KB		8sec	NA	NA	NA	NA
	4KB		15sec	NA	NA	NA	NA
	8KB		26sec	NA	NA	NA	NA
	15KB		63sec	NA	NA	NA	NA
[12]	15KB	AES	3.8sec	27.76	4.2%	NA	NA
		DES	5.07sec	10.13	2%	NA	NA
	30KB	AES	7.5sec	NA	NA	NA	NA
		DES	17.09sec	NA	NA	NA	NA
	75KB	AES	9.33sec	NA	NA	NA	NA
		DES	29.99sec	NA	NA	NA	NA
	90KB	AES	10.7sec	NA	NA	NA	NA
		DES	38.15sec	NA	NA	NA	NA
The Proposed method	3MB	DES	98102 ms	36.70022018 (Byte/ms)	929829KB	15.927	13.40274562
		3DES	341260 ms	10.55021098 (Byte/ms)	1392101KB	97.669	13.40274562
		AES	53353 ms	67.4819598 (Byte/ms)	566144KB	98.378	13.40274562
		BF	1417 ms	2540.83627 (Byte/ms)	692257KB	93.357	13.40274562

[1] - Not Available
[2] - Blowfish algorithm