Computer & Systems Engineering Department (CSED), Spring 2022

Encryption & Decryption

Your data's security, our main priority

Adel Mahmoud, Omar Tarek, Mohammed Hassan Sadek, Mahmoud Attia, Mahmoud Ali, Yahya Azzam
(20010769, 20010998, 20011539, 20011810, 20011811, 20012223)

Abstract

Having the internet reached this level of speed and integrity recently, it became a must for us to find a way to secure our data, one of these ways is encryption. It allows the data to be accessed only by the sender and receiver by transforming the plain text into unreadable ciphertext, a lot of encryption algorithms have been developed in order to build unbreakable encryption, this certain algorithm is defined by the sender.

Keywords:

Encryption, decryption, data security, algorithms, cipher.

1. Introduction:

Data encryption is the process of converting data from its plaintext or readable form into an encrypted one (ciphertext) which converts the text to something like 0's and 1's or random letters and numbers which is harder to be stolen or twisted by anyone. Then use a key to decrypt the message. In symmetric encryption, the same key is used for encryption and decryption, the communicating parties must have the same key in order to achieve secure communication. In asymmetric encryption, each user has two keys: a private key that is known only to the user himself and a public key known to all users. Encryption is done by the public key and the decryption only can be done using the private key that is only known to the receiving party. Encryption requires a mechanism or algorithm which is called a cipher. So, we will list different algorithms, comparing each other, indicating the most efficient algorithms, along with representing flowcharts of at least one algorithm. Then show the benefits and the effects of using encryption techniques on our data to increase security.

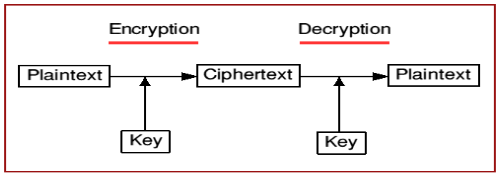


Fig (1)

2. Related work:

The related work to the encryption has various directions, each heading towards different means of encrypting data aiming to reach an optimal solution for stopping the hackers' attacks from succeeding in gaining the information unwanted to be leaked, and as an example the DES, AES, RSA and RC4 algorithms, and even combining two of them together.

a. DES algorithm:

It refers to (Data Encryption Standard) algorithm. It is a symmetric-key block cipher. It is invented in the 1970s by an IBM team. It takes the data as blocks of 64-bit and results in a ciphertext by using a 48-bit key **[5]**. It takes the data in binary (any other forms must be converted to binary) and then divides it into two halves (L0&R0). We do the encryption process in 16 stages to increase security to get (Ln & Rn) such n is from (1) to (16). DES operates with data by 56-bit-keys which are derived from 64-bit. Then create 16-subkey from 1 to 16 from each key after removing some bits to be 48-bits only. Subkeys are the actual participant in encryption. Now we will illustrate the steps with Hammam's video **[8]** help.

Step1: <u>Encoding each 64-bit data block:</u>

The first step is to convert the desired data to binary(64-bit) and then insert them into a permutation matrix (IP) to change the arrangements of data bits starting from the 58th bit ending with the 7th bit. This will result in a 64-permuted block which is divided into L0(left part) and R0(right part) with 32-bit for each. Keys are given and we must convert them to 64-bit binary and decrease them to 56-bit after inserting them into a matrix which removes (the 8th bit and its duplicates). Then each key is divided into two halves Cn&Dn starting with C0&D0 from original key n from 1 to 16 and each Cn &Dn are created from Cn-1&Dn-1 after left shift operations on them but in n=1,2,9,16 we make single shift otherwise we make 2 shifts then insert each result to a matrix to create 48-permuted-bit these are the subkeys (Kn). We now iterate 16 times (1<=n<=16) using a function that operates XOR operation on the 32-bit data block and the 48-bit key Kn to produce blocks of 32-bit then for n from 1 to 16: Ln = Rn-1, Rn=[Ln-1 XOR f (Rn-1, Kn)] where f is the function which operates the XOR operation on Rn-1 and Kn. Briefly, in each iteration, we take the right 32 bits of the previous result and make them the left 32 bits of the current step. For the right 32 bits in the current step, we XOR the left 32 bits of the previous step with the function result as shown in **Fig. (2)**.

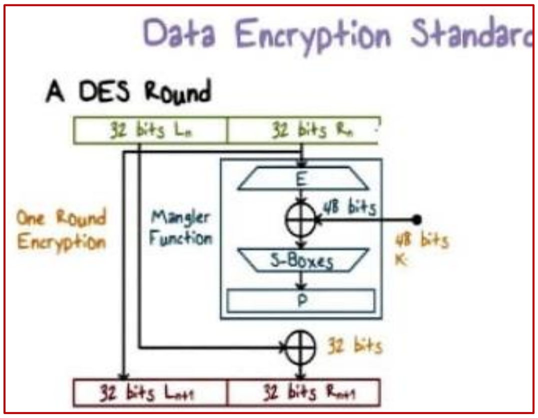


Fig (2)

For example, M the original message M = 1100 1100 0000 0000 1100 1100 1111 1111 1111 0000 1010 1010 1111 0000 1010 1010

L0 = 1100 1100 0000 0000 1100 1100 1111 1111

R0 = 1111 0000 1010 1010 1111 0000 1010 1010

for n=1,

K1 = 000110 110000 001011 101111 111111 000111 000001 110010

L1 = R0 = 1111 0000 1010 1010 1111 0000 1010 1010

R1 = L0 XOR f(R0, K1).

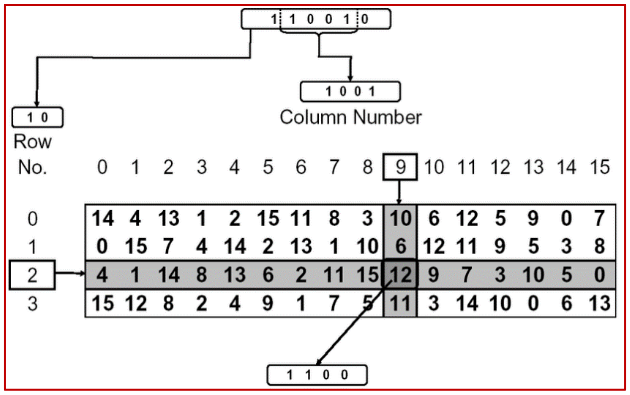
But we have a problem such we can't xor Rn-1 with Kn because they don't have the same number of bits, So, we will expand Rn-1 to be 48-bit by inserting it in E-bit selection table. This matrix(E) will be adding some bits to Rn-1 to be written in 8 blocks of 6 bits each. E(Rn-1) takes a bit from the left and a bit from the right of each 4-bit block of the original Rn-1. For example,

R0 = 1111 0000 1010 1010 1111 0000 1010 1010,

E(R0) = 011110 100001 010101 010101 011110 100001 010101 010101.

After these operations, we can now calculate f(Rn-1, Kn)=E(Rn-1) XOR kn.

Step2: For the operations on the 48-bit result from f we use tables called S-Boxes. The 8 blocks of 6-bits will be used as addresses in different S-boxes S1 to S8 and each block will result in the new address from the corresponding S-box to return a 4-bit one. This will return 32-bit again. This image shows how to return 4-bits from s-boxes. Then the final step in calculating f(Rn-1, Kn) is to insert the resulting 32-bit in a matrix to permute bits.



Row No.	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
0	14	4	13	1	2	15	11	8	3	10	6	12	5	9	0	7
1	0	15	7	4	14	2	13	1	10	6	12	11	9	5	3	8
2	4	1	14	8	13	6	2	11	15	12	9	7	3	10	5	0
3	15	12	8	2	4	9	1	7	5	11	3	14	10	0	6	13

Fig (3)

Step3: After we get all L's to L16 and R's to R16 and swap R16 to be L16 and vice versa. Then insert the new 64-bit into IP^-1 matrix. The resulting message will be well encrypted and increasing the security. And to decrypt this message again to be usable for users we do the same operations but in reverse order. This image describes the previous very well.

b. Advanced Encryption Standard (AES):

Nowadays the Advanced Encryption Standard (AES) has become the most popular and widely used symmetric encryption algorithm. It has been chosen by the U.S. government. The National Institute of Standards and Technology (NIST) started to develop it as a replacement for the DES algorithm whose size was very small **[3]**. AES is six times faster than triple DES. It is used in software and hardware to encrypt sensitive data. It's necessary for the government systems and any systems that need to be protected from any cyberattack. AES is a symmetric block that includes three block ciphers:

- **i.** AES-128 uses a 128-bit key length to encrypt and decrypt a block of messages.
- **ii.** AES-192 uses a 192-bit key length.
- **iii.** AES-256 uses a 256-bit key length.

Each block size is 128 bits. In the 128-bit keys, there are 10 rounds, 12 rounds for the 192-bit keys, and 14 rounds for the 256-bit keys. A round is several processing steps to encrypt the message. So, the more demand of security, the more bits of the system we use. AES provides full specification and design details. Its software is implemented in C and Java. As one of the most factors to consider is the flexibility and suitability for software or hardware implementation.

Description of the algorithm [1]:

1) Key expansion: round keys are derived from the cipher key using the AES key schedule.
2) Initial round key addition: add a round Key where each byte of the state is combined with a byte of the round key using bitwise XOR.
3) 9, 11, or 13 rounds:
- Sub Bytes: a non-linear substitution step where each byte is replaced with another according to a lookup table.
- Shift Rows: a transposition step where the last three rows of the state are shifted cyclically for a certain number of steps.
- Mix Columns: a linear mixing operation that operates on the columns of the state, combining the four bytes in each column.
- Add round Key
4) Final round:
    - Sub Bytes
    - Shift Rows
    - Add Round Key
    - After this step there will be 10, 12, or 14 round

c. RSA algorithm [4]:

It comes from the surnames Ron Rivest, Adi Shamir, and Leonard Adleman. It is asymmetric encryption that needs two keys, public and private keys for encryption and decryption. To start up with how actually this algorithm works, I'll introduce a simple example but first, we need to know that the encryption and decryption keys consist of two pairs one for each by which we can encrypt and decrypt whatever we want let's dig into this example right away:

Suppose we want to send a message which contains a single letter, say 'B', we send it as a number, of course, say it's 2, then we have to decrypt this number, using the decryption function with the help of the two entries of the encryption pair suppose it's (5, 14), the function is as follows: (2^5)%14,

This gives us 4 whose letter equivalent is 'D', and this's the encrypted message that will be sent, now the receiver has got his secret key for decryption that consists also

of a pair of numbers whose second entry is the same as the one of the encryption pair! Then how is it secret? Well, the 1st entry is not known and that's what makes it secret. Now say that the decryption pair is (11, 14), the receiver has got a message which is 'D', and he wants to decrypt it using the decryption function which is: (4^11)% 4, which gives 2 back again whose letter equivalent is 'B', which's the desired message. Now we're done with our simple example, let's see how we've created the encryption and decryption pairs:

I. We choose two enormous prime numbers p and q.
II. After that we multiply them to get a number call it N, and this is the 2nd entry for both encryption and decryption pairs.
III. Then we get the number of the co-primes of N by using the PHY function which is as follows: f(N) = (p – 1) (q – 1).
IV. then we get the 1st entry of encryption call it 'e', such that it obeys two properties:
    - lies between 1 and f (N).
    - must be coprime with N and f (N).
V. then we get the 1st entry of decryption to call it 'd' such that: d*e % f (N) = 1.

Now we've successfully built the encryption pair (the lock), and the decryption pair (the key).

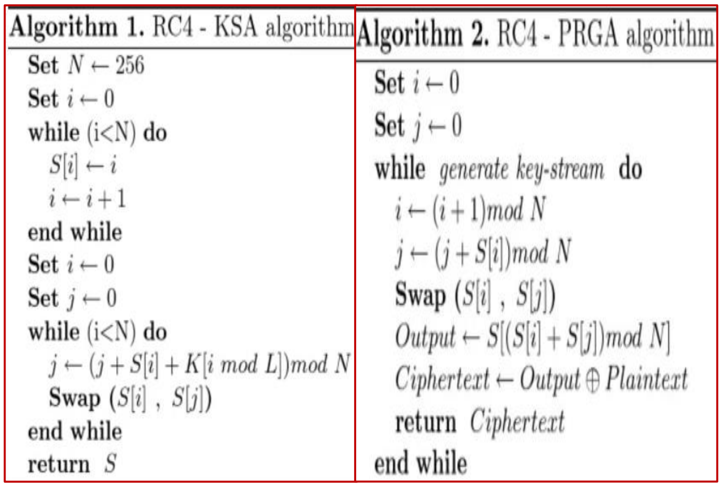
Note: if we apply the previous steps to our simple example by choosing p and q as 2, and 7 we'll get our message encrypted and decrypted successfully.

d. RC4 algorithm [2]:

The RC4 algorithm was designed in 1987 by Ronald Rivest secretly until it was disclosed in 1994. It uses a symmetric key like the DES algorithm but RC4 is better than DES in terms of speed and complexity. It has many uses and applications like Wi-Fi protected access (WPs), Transport Layer Security (TLS) in web browsing and emailing and messaging security.

Description of the algorithm: The RC4 algorithm consists of two mathematical operations:

1) KSA: a Key Scheduling Algorithm to start the process in an array (typically called "S") which is processed 256 times and the bytes of the key are mixed in it. The KSA generates initial switching by mixing the corresponding switch using the key and this is the entrance of the PRGA part. The secret key produces random swapping as the main idea of the RC4 algorithm is making the switching of the elements by exchanging them for higher randomization as shown in algorithm1.
2) PRGA: Pseudo-Random Generation Algorithm which generates bytes of the keystream which is used to perform XOR operation with the plaintext to get the ciphertext as shown in **Fig. (4), Algorithm2.**



Algorithm 1. RC4 - KSA algorithm

```
Set $N \leftarrow 256$
Set $i \leftarrow 0$
while (i<N) do
  $S[i] \leftarrow i$
  $i \leftarrow i+1$
end while
Set $i \leftarrow 0$
Set $j \leftarrow 0$
while (i<N) do
  $j \leftarrow (j + S[i] + K[i \ mod \ L])mod \ N$
  Swap $(S[i] \ , \ S[j])$
end while
return $S$
```

Algorithm 2. RC4 - PRGA algorithm

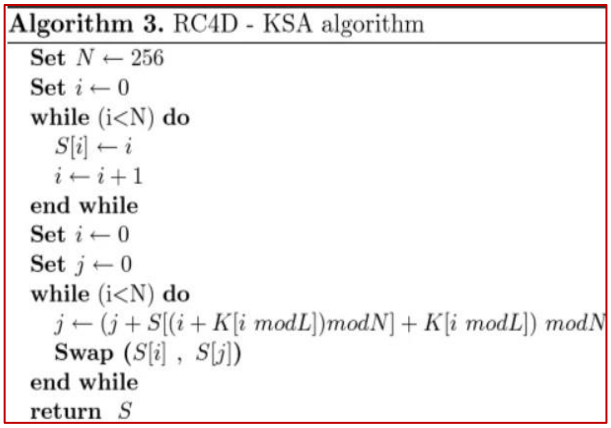
```
Set $i \leftarrow 0$
Set $j \leftarrow 0$
while $generate \ key\text{-}stream$ do
  $i \leftarrow (i+1)mod \ N$
  $j \leftarrow (j + S[i])mod \ N$
  Swap $(S[i] \ , \ S[j])$
  $Output \leftarrow S[(S[i] + S[j])mod \ N]$
  $Ciphertext \leftarrow Output \oplus Plaintext$
  return $Ciphertext$
end while
```

Fig (4)

As the RC4 algorithm has many vulnerabilities caused by cryptanalysis of its key and breaking encrypted messages, there was a great need to modify and improve it. So, it was modified and called (RC4D) by modifying its two parts:

1) The KSA was modified by adding a new random variable to obtain higher randomization as shown in algorithm 3.
2) The PRGA was modified by producing a new random variable and performing XOR operation on the new random variable and the j random variable which is used to access the data elements as shown in the figure below then performing XOR operation on its output and the plaintext to get the ciphertext as shown in algorithm 4.

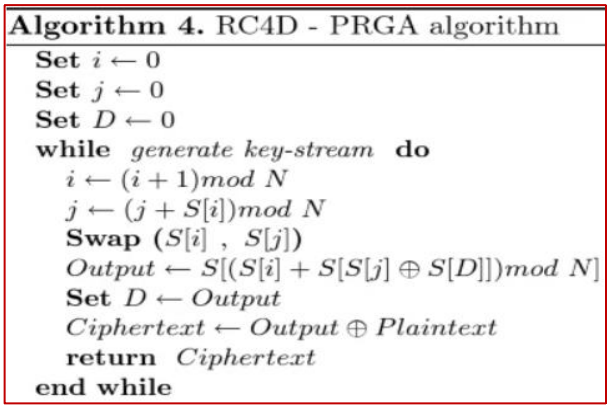
This modification increased the degree of randomness and improved this algorithm without taking much time as it takes places 265 times only.



Algorithm 3. RC4D - KSA algorithm

```
Set $N \leftarrow 256$
Set $i \leftarrow 0$
while (i<N) do
  $S[i] \leftarrow i$
  $i \leftarrow i+1$
end while
Set $i \leftarrow 0$
Set $j \leftarrow 0$
while (i<N) do
  $j \leftarrow (j + S[(i + K[i \ modL])modN] + K[i \ modL]) \ modN$
  Swap $(S[i] \ , \ S[j])$
end while
return $S$
```

Fig (5)



Algorithm 4. RC4D - PRGA algorithm

```
Set $i \leftarrow 0$
Set $j \leftarrow 0$
Set $D \leftarrow 0$
while $generate \ key\text{-}stream$ do
  $i \leftarrow (i+1)mod \ N$
  $j \leftarrow (j + S[i])mod \ N$
  Swap $(S[i] \ , \ S[j])$
  $Output \leftarrow S[(S[i] + S[S[j] \oplus S[D]])mod \ N]$
  Set $D \leftarrow Output$
  $Ciphertext \leftarrow Output \oplus Plaintext$
  return $Ciphertext$
end while
```

Fig (6)

e. Mixed Technique of Data Encryption Between DES and RC4 Algorithm [6]:

Because the DES algorithm and RC4 algorithm can be broken and hacked later we have tried to study a new algorithm that is designed to combine the advantages of both DES & RC4 to overcome their deficiencies and disadvantages.

The "DES" algorithm encrypts a block of 64-bit by separating it into two segments left and right where each part takes 32bit. By this effort, the left side of the "DES" algorithm acts according to its own principle, while the right one simply acts as an "RC4" algorithm. This gives the algorithm more complexity and makes it more difficult to break. The key is inserted in the RC4 algorithm located at the right part i.e., we execute an "XOR" operation on the message bits along with the key to attain the second key, then insert it in the RC4 algorithm again and continue until we complete all stages of DES algorithm successfully. Thus, the resulting code will be very complex and hard to decrypt.

The flowchart in **Fig. (7)** is showing the combination process between (DES & RC4) algorithms cipher with the DES algorithm working by Block cipher and **Fig. (8)** is showing how to insert the RC4 algorithm working by stream.

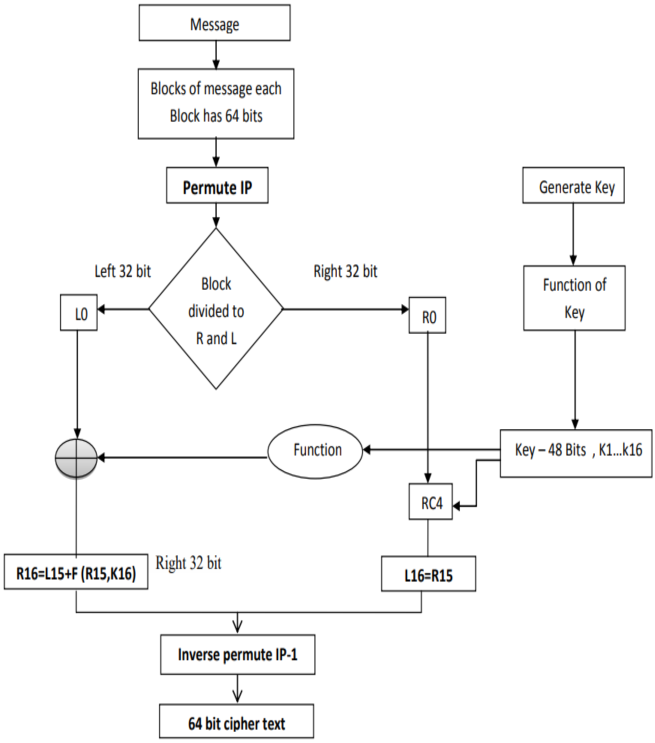


Fig (7)

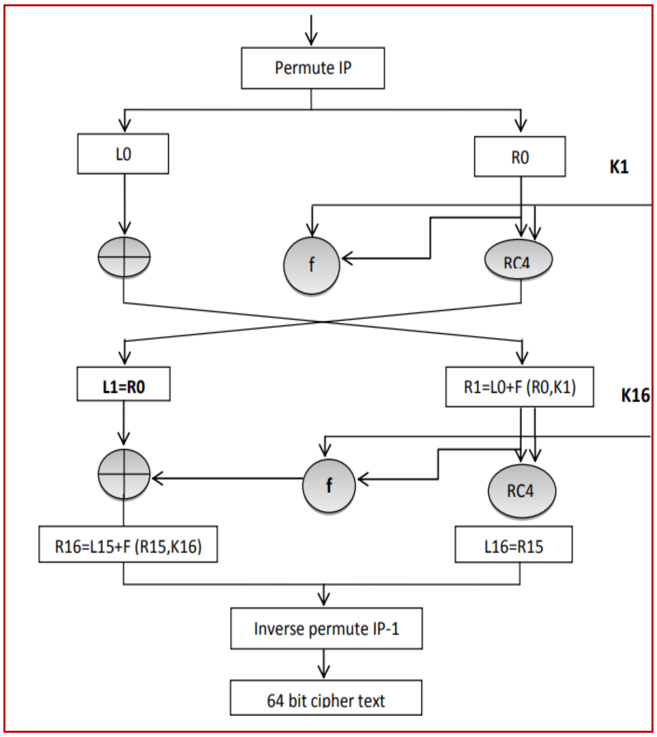


Fig (8)

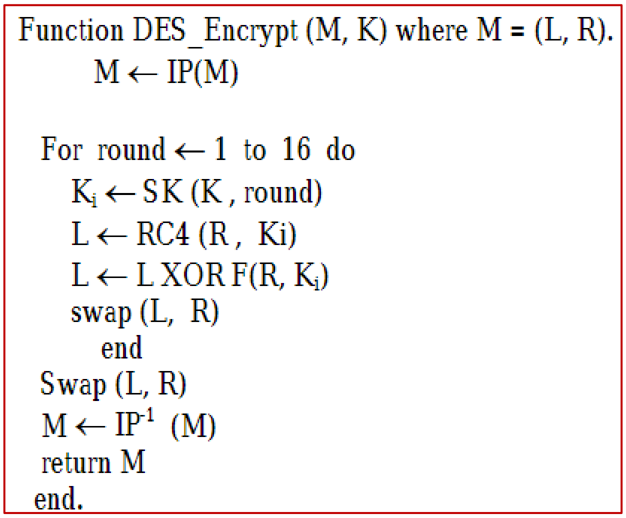
The algorithm of the following flowchart is:

Pseudocode:

1) Message.
2) Convert the message to HEX (base 16).
3) Convert Hex to Bin where the base is 64-bit.

Processing: The following pseudo-code shows the algorithm of encryption and decryption, and we can see that they are similar, and the only difference is the order in which the keys are used:

1) Encryption algorithm:

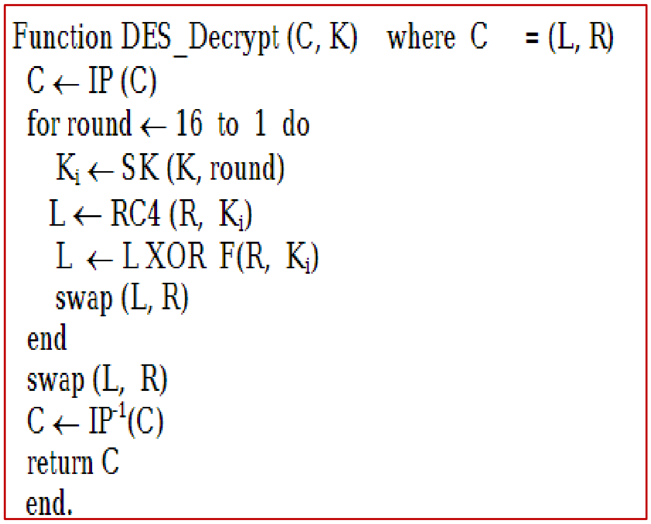


```
Function DES_Encrypt (M, K) where M = (L, R).
      M ← IP(M)

  For round ← 1 to 16 do
    $K_i$ ← SK (K , round)
    L ← RC4 (R , Ki)
    L ← L XOR F(R, $K_i$)
    swap (L, R)
      end
  Swap (L, R)
  M ← $IP^{-1}$ (M)
  return M
 end.
```

Fig (9)

2) Decryption algorithm:



```
Function DES_Decrypt (C, K)   where C   = (L, R)
 C ← IP (C)
 for round ← 16 to 1 do
    $K_i$ ← SK (K, round)
   L ← RC4 (R, $K_j$)
    L ← L XOR F(R, $K_j$)
    swap (L, R)
 end
 swap (L, R)
 C ← $IP^{-1}$(C)
 return C
 end.
```

Fig (10)

- **A comparison between the new algorithm and the DES algorithm after applied using "visual Basic.net 2010" is given in the following table:**

Table (1) [6]

Number of blocks	DES only	DES & RC4	Number of blocks	DES only	DES & RC4
4	0.00034	0.00042	128	0.0068	0.0088
8	0.00062	0.00078	256	0.0118	0.0159
16	0.0012	0.0016	512	0.023	0.0297
32	0.002	0.0027	1024	0.0406	0.05
64	0.0034	0.0049			

Then the complexity of the DES algorithm is computed based on the length of text required to be encoded:

Complexity = (64! * 16 Round)

However, to compute the complexity of the new algorithm:

Complexity = (64! * 32! * 32 Round) *16 Round

Where:

- (64!): represents the length of the block required to be encoded.
- (32!): represents the size of half of the block entered from the right-hand side using the RC4 algorithm.

And the following chart clarifies the time difference between the new algorithm and the DES algorithm:

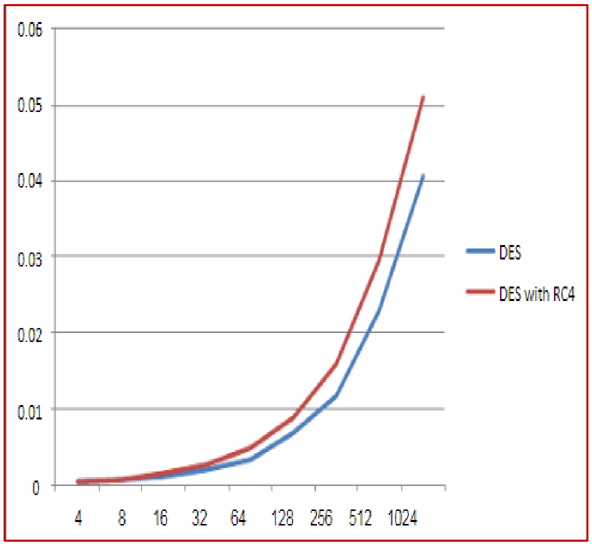


Fig (11)

So, we can see that the new algorithm takes a little greater time than the DES algorithm, but the new algorithm provides us with a high degree of security and increases the proportion of the complexity of the codebreaking.

3. Methods:

From the methods mentioned, we can make a quick comparison between the different methods:

1) DES encryption algorithm:

Short for **D**ata **E**ncryption **S**tandard. It's one of the earliest algorithms. Now, DES is considered insecure due to its small key size (56 bits). But triple DES algorithm applies DES algorithm three times so it's better in terms of security as it has 3 keys each of 56 bits. DES is a symmetric-key block cipher.

2) AES encryption algorithm:

Short for **A**dvanced **E**ncryption **S**tandard. AES is the successor to DES where DES got cracked so the AES took it place since 2003, and it got some competitors like RC4. AES is based on Rijndael cipher. AES is a symmetric-key block cipher. Now most CPUs include hardware AES support making it very fast. AES is supported by TrueCrypt, SSH. AES has many modes (AES-GCM, AES-CBC and others)

3) RSA encryption algorithm:

Short for the surnames of its designers Ron **R**ivest, Adi **S**hamir and Leonard **A**dleman. RSA is not used directly to encrypt because of speed constraints and its yield is small and usually it's used to share a secret key after that, a symmetric key algorithm is used for the actual encryption.

4) RC4 encryption algorithm:

Short for **R**ivest **C**ipher 4 or **R**on's **C**ode 4. It uses a symmetric key. RC4 is a stream cipher as recently block cipher were found to have some attacks for example BEAST, Lucky13. And RC4 is recommended against those attacks as it uses a stream cipher.

4. Results:

Throughout the ages experts tried to reach new encryption algorithms that ensure a high degree of security and too hard to break. From our discussion and research, we have found that each time they reach a new encryption algorithm they found some defects in it so they tried to find another one that handles those defects of the previous one. But unfortunately, they found that the new one has some defects as well. For example, RC4 algorithm solved many defects of the DES algorithm as RC4 algorithm was faster and more secure than DES algorithm but RC4 algorithm itself has some vulnerabilities that enable attackers to break the encrypted messages without the known cryptographic key. So, we found that if we tried to combine two algorithms together to get the benefits of each one in one algorithm, we will reach to a new algorithm that is in an extremely high degree of security. For example, as we discussed in section (2) part (e) when combining RC4 algorithm with DES algorithm we get a new algorithm that provides us with a high degree of security and increases the proportion of the complexity of the codebreaking. And we think that the idea of combining two algorithms in one new algorithm will be the future of encryption algorithms.

5. Conclusion:

Cryptography has played a vital role in keeping our data safe from hacking and other threats. Cryptographic algorithms have been developed over the years starting with Caesar Cipher (invented around 100 BC) **[7]**, which was simply by shifting each letter 3 steps back like that:

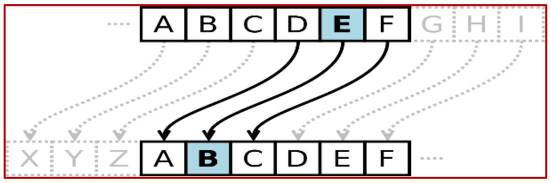


Fig (12)

up till 2005 when ECC (Elliptic-Curve-cryptography) has been invented, and you may ask yourself why all of that? The answer is, the more cryptanalysis evolves, the more powerful algorithm we'll need for the purpose of securing all kinds of data – personal, financial, medical, commercial, etc.., so each following algorithm is trying to eliminate or decrease the vulnerabilities of its previous one to build one that's much harder to break. we've presented a review of some public algorithms - DES, AES, RSA, and RC4 for each we've explained how encryption and decryption process work in detail, and how it contributes to better security than the previous one.

6. References:

[1] S. A. Loay, D. Yehia, and J. Mohammed. (2020). AES Encryption: Study & Evaluation. [Book]. Available: (PDF) AES Encryption: Study & Evaluation (researchgate.net)

[2] R.Alsharida, M.Hammood, M.A.Ahmed, B. Thamer, M.Shakir, (2021). Presented at 12th Int. Networking Conf. (INC 2020) [Online], Available: RC4D: A New Development of RC4 Encryption Algorithm | SpringerLink

[3] C. Bernstein, M. Cobb. "Advanced Encryption Standard (AES)" techtarget.com What is the Advanced Encryption Standard (AES)? Definition from SearchSecurity (techtarget.com) (recessed Apr. 29, 2022)

[4] J. Lake. "What is RSA encryption and how does it work?" What is RSA encryption and how does it work? (comparitech.com) (recessed Apr. 29, 2022)

[5] Simplilearn. "What is DES (Data Encryption Standard): DES Algorithm and Operation. What Is DES (Data Encryption Standard): DES Algorithm and Operation [Updated] (simplilearn.com) (recessed Apr. 29, 2022)

[6] T. Hameed "A Robust Approach for Mixed Technique of Data Encryption Between DES and RC4 Algorithm" in ResearchGate, Mar 2022. [Online]. Available: (PDF) A Robust Approach for Mixed Technique of Data Encryption Between DES and RC4 Algorithm (researchgate.net)

[7] Khan Academy, Mia Epner, U.K. Encryption and public keys | Internet 101 | Computer Science | Khan Academy. (Apr. 23, 2019). Accessed: Apr. 29, 2022. [Online Video]. Available: Encryption and public keys | Internet 101 | Computer Science | Khan Academy - YouTube

[8] Student Guide دليل الطالب, Islam Hammam, Egypt. DES algorithm شرح مبسط كامل التشفير و فك التشفير , S BOX , Data Encryption Standard. (Jan. 17, 2021). Accessed Apr. 29, 2022. [Online Video]. Available: DES algorithm شرح مبسط كامل التشفير و فك التشفير , S BOX , Data Encryption Standard - YouTube

7. Plagiarism percentage: 10%