THE EFFICIENCY OF ENCRYPTION ALGORITHMS IN ENCRYPTING LARGE FILES

Mohammed Abdel Lateif Al-Shalabi

Department of Information Technology, Sharjah University

Muath Maqableh

Department of Information Technology, Hael University

Alaa Obeidat

Department of Information Technology, Sharjah University

ABSTRACT: In this paper, we implemented five of the most popular and used encryption algorithms in the world, AES, DES, RC4, RC5, and RSA.

We made a comparison among these algorithms on the same OS, and another comparison between the same algorithms on different OSs, these comparisons is the term of time performance.

The idea behind this project is the importance of security nowadays, especially when the Internet became widely used by millions of peoples for secured transactions, so, these transactions may contain sensitive data like Password, Credit Card Number, or others, these sensitive data must be transmitted over a secure channel to avoid any attack on it.

Also, the thing which is differentiate our research is using a combined file of both Arabic and English symbols, the encryption and decryption processes for Arabic symbols require more additional code and algorithm to deal with Arabic symbols.

KEYWORD: Cryptography, Security Algorithms, DES, RSA, RC4.

1. INTRODUCTION

One of the most common uses of encryption is in electronic messaging. Encryption can be used to secure email on public and private networks. Unlike e-mail on a private system, which goes directly to a mail server and resides there until it is retrieved, Internet e-mail bounces from server to server on its way to a recipient. This makes the transmission channel impossible to secure and provides numerous opportunities for interception. Here it makes sense to secure the message itself by using encryption. But private networks are not immune to the need for higher security and often employ encryption to guarantee the integrity of the message.[***1]

2. EXPERIMENTAL DESIGN

This study is conducted for different popular encryption algorithms such as DES, AES, RC4, RC5, and RSA, they were implemented and their performance was compared by encrypting large files (1 Mb) which consist of both Arabic and English symbols.

The algorithms were implemented using JAVA programming language because it supports many useful packages and classes, and they had been run on Windows XP operating system with CPU speed of 500Mhz .

A comparison is conducted between the results of the selected algorithms in terms of the encryption time to produce a cipher text, also the study is performed on the selected encryption algorithms after modifying them in the encryption process to increase the efficiency of the cipher text, we mean that we made a double encryption, the first one is the encryption process of the algorithm itself, and the other is the added code from our modification.

Also, the another thing which is differentiate our research is using a combined file of both Arabic and English symbols, and the encryption/decryption processes for Arabic symbols require more additional code and algorithm to deal with Arabic symbols.

3. RESEARCH METHOD

a. AES Algorithm

The most common way to attack block ciphers is to try various attacks on versions of the cipher with a reduced number of rounds. AES has 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. The best-known attacks are on 7 rounds for 128-bit keys, 8 rounds for 192-bit keys, and 9 rounds for 256-bit keys. [***3]

AES is designed to work on bytes. However, each byte is interpreted as a representation of the polynomial: [***4]

$$b_7x_7 + b_6x_6 + b_5x_5 + b_4x_4 + b_3x_3 + b_2x_2 + b_1x_1 + b_0x_0$$

where each bi is either 0 or 1.[***5]

b. DES Algorithm

The following notation is convenient: Given two blocks L and R of bits.[***6]

c. RC4 Algorithm

RC4 is asymmetric encryption algorithm that uses a single key for both encryption and decryption. This algorithm works by uses the concept of substitution box named as S-box (one dimension array that range from 0 to 255. [***7]

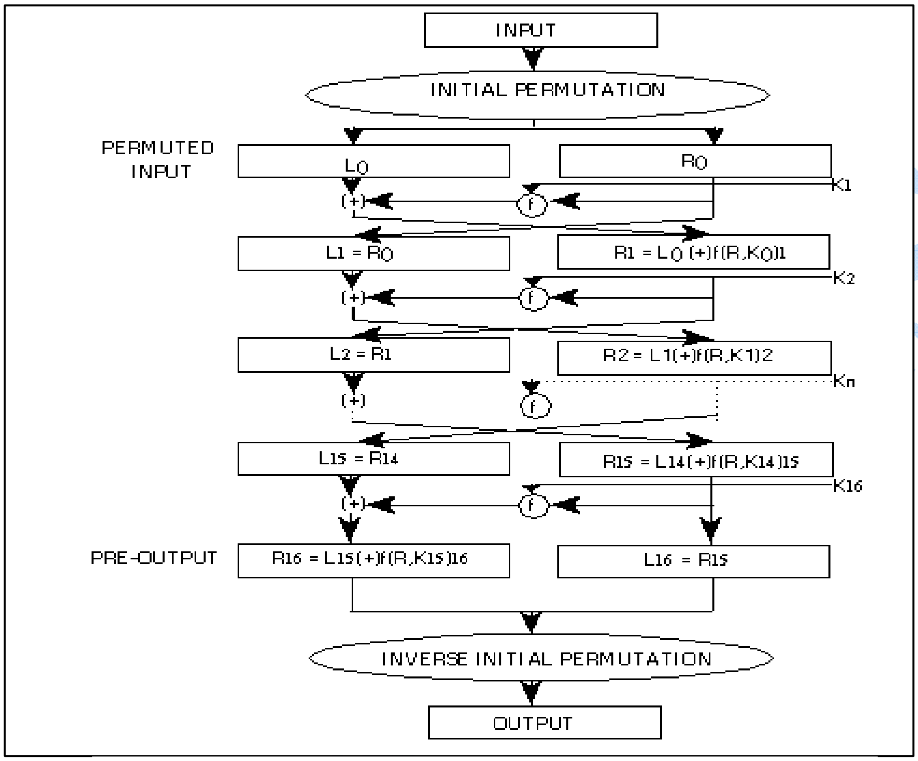


Figure 1. Enciphering Computation

d. RC5 Algorithm

In this section we describe the RC5 algorithm, which consists of three Components, a key expansion algorithm, an encryption algorithm, and a decryption Algorithm. [***2] We assume that the input block is given in two w_bit registers A and B, we also assume 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]

```
A = A + S [0];
B = B + S [1];
For i =1 to r do
A =((A XOR B) <<< B) + S [2*i];
B =((B XOR A) <<< A) + S [2*i+1];
```

The output is in the registers A and B.[***8]

e. RSA Algorithm

The RSA algorithm was described in 1977 by Ron Rivest, Adi Shamir, and Len Adleman who were all at MIT at a time. [***A] RSA depends on the one-way function, means that the encryption and decryption are done in separate functions, and uses large integers (e.g. 1024 bits), so, the security of the RSA depends on the cost of factoring large numbers [***A]

4. RESULTS AND ANALYSIS

In this section we will discuss the performance of the five encryption algorithms in the term of the elapsed time for the encryption process.

4.1 Performance

We implemented these algorithms using JAVA programming languages, because it supports many useful packages and classes, each encryption algorithm in separate class to handle the comparison aims, and we use the same test data (1 MB text file) for all the classes (algorithms) and records the results, especially the elapsed time for each encryption algorithm, now we will see the results after running the algorithms on Windows XP Operating System with CPU speed of 500Mhz, but the comparisons we will see in the next section.

In the figure 2, we see the GUI that the user contacts with. The interface consists of one Text Area, and four buttons (Open, Encrypt, Decrypt, and Clear), now when a user wants to encrypt a file using any encryption algorithm, he/she must click the Open button, after that, the Open Dialog as shown in figure 3 will appear to allow the user selects the file he/she wants, then the contents of the file will appear in the text area, after that, the user must click the Encrypt button to apply the encryption process, and the encrypted data will appear in the text area with the elapsed time consuming for this process.

The user must apply the same procedure when he/she wants to decrypt the encrypted data, he/she must click the Decrypt button, and then the decrypted data will appear in the text area with the elapsed time consuming for the decryption process.

We used these elapsed times to make a comparison among the different encryption algorithms, as we will see later.

Figure 4 shows the encryption processes for the *AES encryption algorithm* with the elapsed time for the process. Figure 5 shows the encryption processes for the *DES encryption algorithm* with the elapsed time for the process. Figure 6 shows the encryption processes for the *RC4 encryption* algorithm with the elapsed time for the process. Figure 7 shows the encryption processes for the *RC5 encryption* algorithm with the elapsed time for the process. Figure 8 shows the encryption processes for the *RSA encryption* algorithm with the elapsed time for the process.

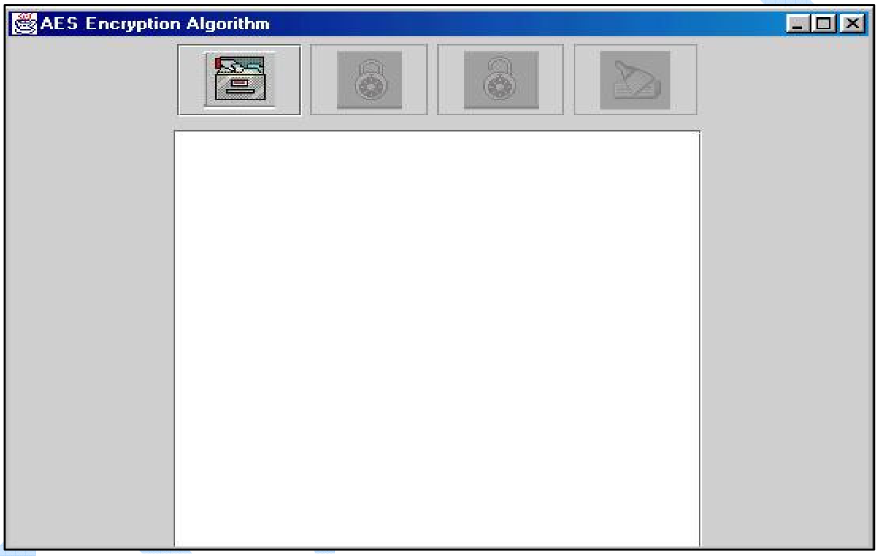
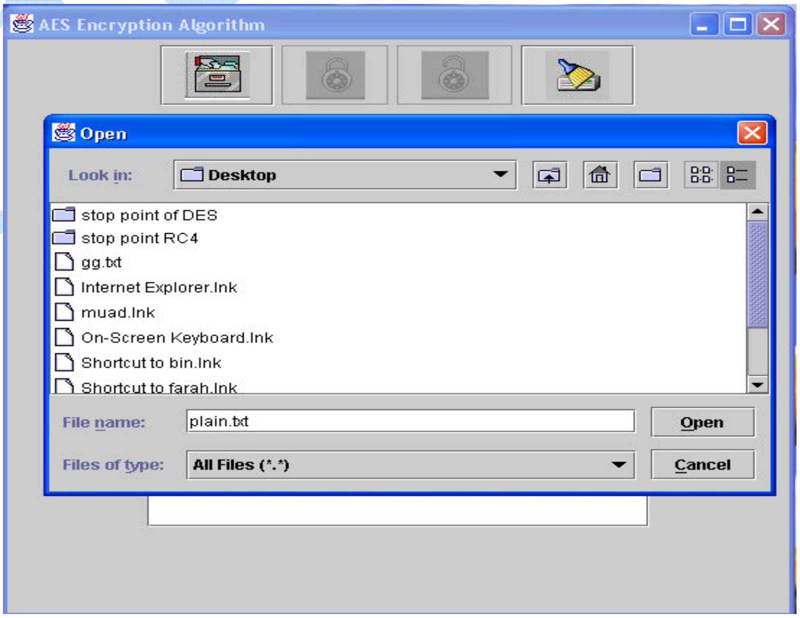


Figure 2. The Main Screen



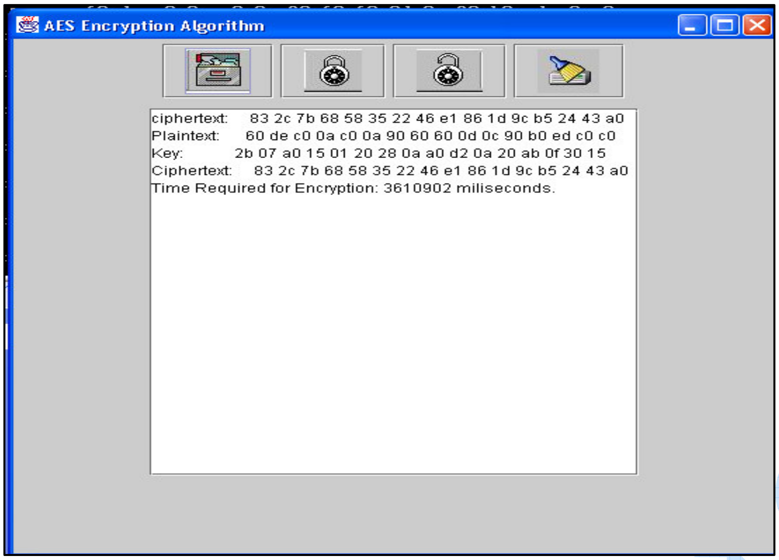
Look in: Desktop

stop point of DES
stop point RC4
gg.txt
Internet Explorer.lnk
muad.lnk
On-Screen Keyboard.lnk
Shortcut to bin.lnk
Shortcut to farah.lnk

File name: plain.txt

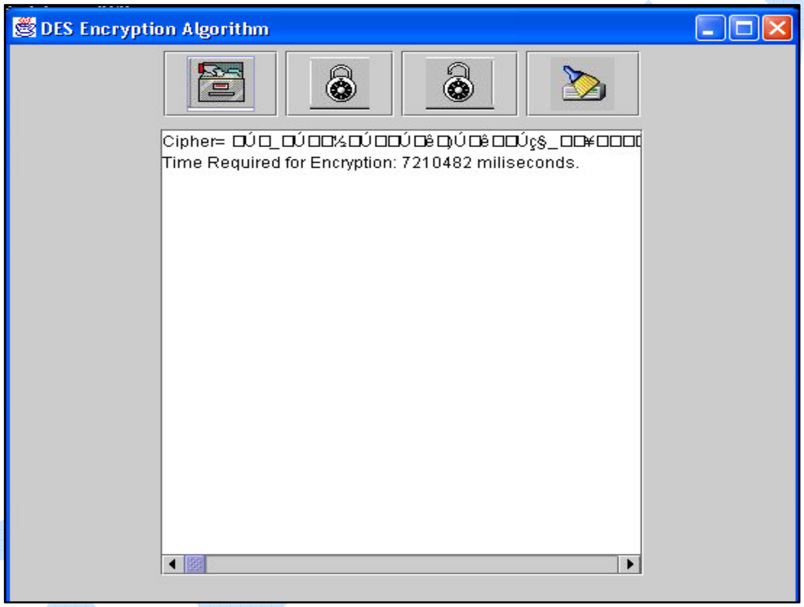
Files of type: All Files (*.*)

Figure 3. The Open Dialog



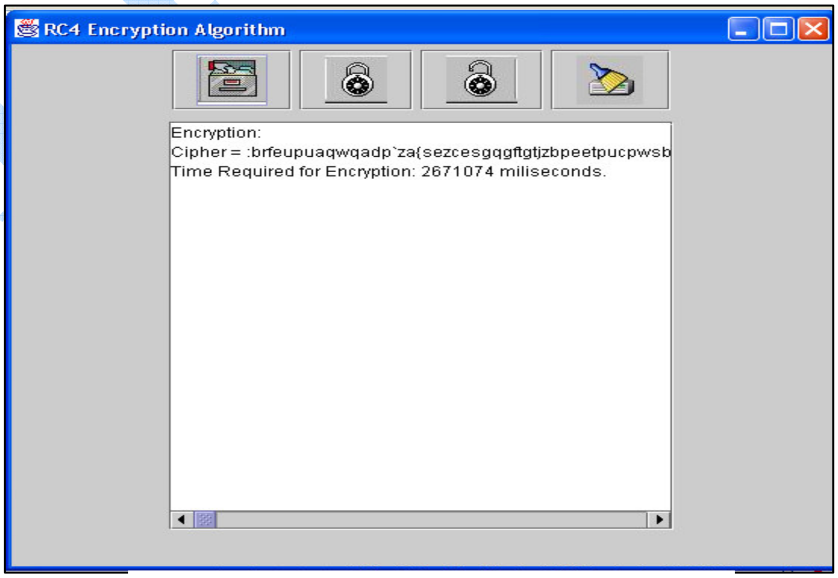
```
ciphertext:    83 2c 7b 68 58 35 22 46 e1 86 1d 9c b5 24 43 a0
Plaintext:     60 de c0 0a c0 0a 90 60 60 0d 0c 90 b0 ed c0 c0
Key:           2b 07 a0 15 01 20 28 0a a0 d2 0a 20 ab 0f 30 15
Ciphertext:    83 2c 7b 68 58 35 22 46 e1 86 1d 9c b5 24 43 a0
Time Required for Encryption: 3610902 miliseconds.
```

Figure 4. The Encryption Process



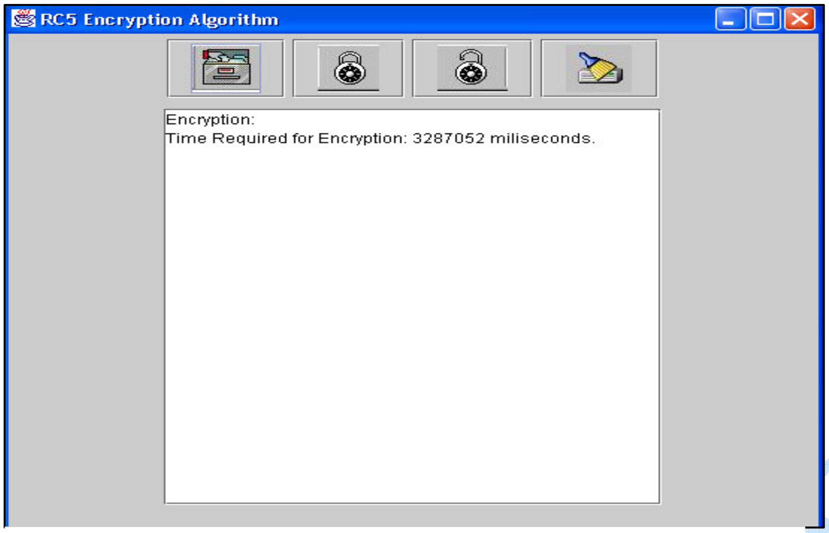
```
Cipher= □Ú□_□Ú□□½□Ú□□Ú□ê□Ú□ê□□Úç§_□□¥□□□□
Time Required for Encryption: 7210482 miliseconds.
```

Figure 5. The Encryption Process



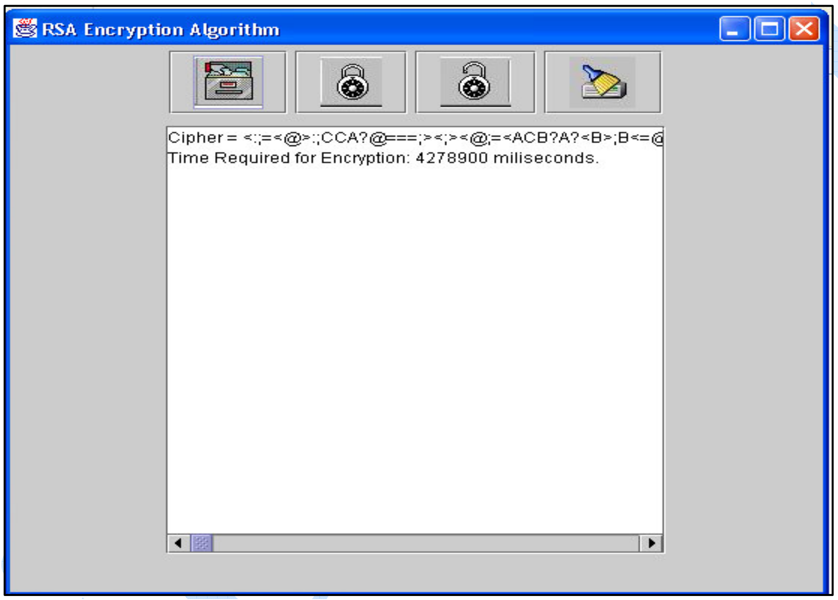
```
Encryption:
Cipher = :brfeupuaqwqadp`za{sezcesgqgftgtjzbpeetpucpwsb
Time Required for Encryption: 2671074 miliseconds.
```

Figure 6. The Encryption Process



```
Encryption:
Time Required for Encryption: 3287052 miliseconds.
```

Figure 7. The Encryption Process



```
Cipher = <:;=<@>:;CCA?@===;><;><@;=<ACB?A?<B>;B<=@
Time Required for Encryption: 4278900 miliseconds.
```

Figure 8. The Encryption Process

4.2 Analysis

As we saw in the previous section, the elapsed time differs from one encryption algorithm to another, according to nature of the algorithms as we discussed before, for example, and by a logic, the DES algorithm has the highest elapsed time because it uses 16 rounds, and in each round, there are many functions to be computed like Key Expansion, and others, then RSA algorithm, because it applied a mod function and large key's size, then the AES algorithm, because it has only 10 rounds, and each round has four functions, RC5 algorithm has less elapsed time, because it has many logical operations like Shift Left, Shift Right, XOR, and others, finally, the RC4 algorithm, because it has small operations like Swap and XOR functions.

The results of comparison are represented using charts, as explained in the figure 9 below:

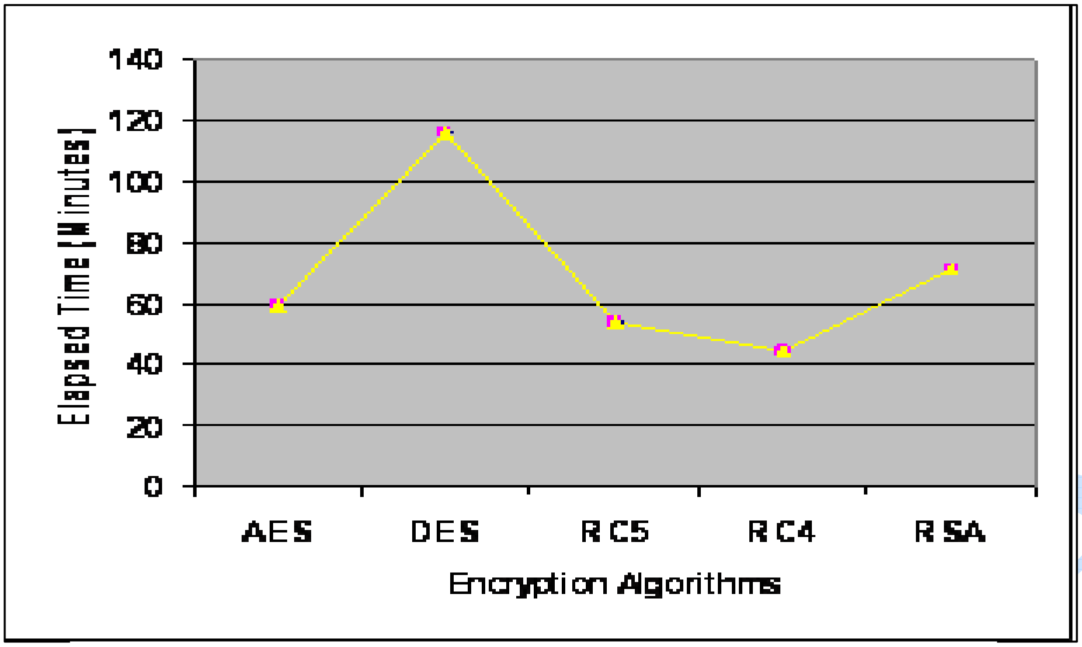


Figure 9. The Comparison Chart

CONCLUSION

The presented results showed that DES algorithm has the highest elapsed time, since DES uses 16 rounds, and there are many functions to be computed like Key Expansion, and others, then RSA algorithm, because it applied a *Mod* function and large key's size, then the AES algorithm, because it has only 10 rounds, RC5 algorithm has less elapsed time, because it has many logical operations like Shift Left, Shift Right, XOR, and others, finally, the RC4 algorithm, because it has small operations like Swap and XOR functions**,** the results also based on using large files and some Arabic symbols in the file, and also we made some modifications on the algorithms to increase the security of them.

ACKNOWLEDGEMENTS

The authors of this study would like to thank the reviewers for their valuable comments to improve the representation of this paper.

REFERENCES

[***1] http://www.verisign.com

[***2] http://www.mycrypto.net

[***3] http://portal.acm.org

[***4] http://www.w3.org

[***5] http://www.minrank.org/aes/

[***6] http://bass.gmu.edu

[***7] http://www.aci.net

[***8] http://www.wisdom.weizmann.ac.il

[***9] http://www.comms.scitech.susx.ac.uk

[***A] http://slwww.epfl.ch/CA/rsa/