Efficiency of AES finalist candidate algorithms

Hanna Rasheed Ismaeel

Al-Nahrain University

Abstract:

The Advanced Encryption Standard (AES) are one of the most algorithms used in symmetric key cryptography. Finalist candidate algorithms are five AES algorithms they are :MARS, RC6, Rijndael , Serpent, and Twofish. The paper evaluate three algorithms which are Rijndael, Serpent, and Twofish. The reason of chosen these three algorithms are :they have fixed block size of 128 bits and a key size of 128, 192, or 256 bits. We implement these algorithms in two kinds of computers: desktop pc. (Samsung 3GB) and laptop pc. (Intel core2 2GB).The time required for decryption and encryption Are Measured and the results are compared for the three algorithms we find that Rijndael is the best.

1-introduction

The National Institute of Standards and Technology (NIST) has been working with the international cryptographic community to develop an Advanced Encryption Standard (AES). The overall goal is to develop a Federal Information Processing Standard (FIPS) that specifies an encryption algorithm capable of protecting sensitive (unclassified) government information well into the twenty-first century. NIST expects that the algorithm will be used by the U.S. Government and, on a voluntary basis, by the private sector.In 1997, NIST initiated a process to select a symmetric-key encryption algorithm to be used to protect sensitive (unclassified) In 1998, NIST announced the acceptance of fifteen candidate algorithms and requested the assistance of the cryptographic research community in analyzing the candidates. This analysis included an initial examination of the security and efficiency characteristics for each algorithm. NIST reviewed the results of this preliminary research and selected MARS, RC6, Rijndael, Serpent and Twofish as finalists [1].

2-The Aim of the Research :

The basic aim of the research is to measure the efficiency of AES algorithm by implement it in different platform with different size of memory and a processor ,then make a comparison between the results.

3-Outline of the research

1-Description of the AES.

2-Define the goal of AES

3-Define the evaluation criteria of AES algorithm.

4-Measure the performance of (Rijndal,towfish and serpent)

5-implement the three algorithm in different platforms.

6-make a comparisons between the results.

4-Description of the AES :

3.1 -AES has a fixed block size of 128 bits and a key size of 128, 192, or 256 bits.

3.2 Rijndael can be specified with block and key sizes in any multiple of 32 bits, with a minimum of 128 bits and a maximum of 256 bits. Assuming one byte equals 8 bits, the fixed block size of 128 bits is 128 ÷ 8 = 16 bytes.[2]

3.3 AES operates on a 4×4 array of bytes, Most AES calculations are done in a special finite field. the AES cipher is specified as a number of repetitions of transformation rounds that convert the input plain-text into the final output of cipher-text. Each round consists of several processing steps, including one that depends on the encryption key. A set of reverse rounds are applied to transform cipher-text back into the original plain-text using the same encryption key.

5-Goals Of AES

• Very strong symmetric block cipher for government and commercial use in the next century[3]

• More efficient than Triple DES

• More secure than Triple DES

– Key sizes: 128, 192, and 256 bits

– Block sizes: 128 bits (other sizes optional)

• Publicly defined and evaluated

• Worldwide royalty free

6-Evaluation Criteria of AES Algorithm:

NIST specified the overall evaluation criteria that would be used to compare the candidate algorithms. The evaluation criteria were divided into three major categories [4]

1-Security: was the most important factor in the evaluation and encompassed features such as resistance of the algorithm to cryptanalysis, soundness of its mathematical basis, randomness of the algorithm output, and relative security as compared to other candidates.

2-Cost:A second important area of evaluation was cost that encompassed licensing requirements, computational efficiency (speed) on various platforms, and memory requirements .The speed of the algorithm on a variety of platforms needed to be considered. During Round 1, the focus was primarily on the speed associated with 128-bit keys. During Round 2, hardware implementations and the speeds associated with the 192 and 256-bit key sizes were addressed. Memory requirements and software implementation constraints for software implementations of the candidates were also important considerations.[5]

3-Algorithm and Implementation Characteristics.: such as flexibility which includes the ability of an algorithm to handle key and block sizes beyond the minimum that must be supported ,And to be implemented securely and efficiently in many different types of environments , and to provide additional cryptographic services.

7-AES Finalist Algorithm

7.1 The characteristics of Rijndael Algorithm:

1. **The block cipher Rijndael was designed by Joan Daemen and Vincent Rijmen as a candidate for the Advanced Encryption Standard. [6]**
2. **The algorithm can be implemented very efficiently on a wide range of processors and in hardware.**
3. **Rijndael's key length is defined to be either 128, 192, or 256 bits in accordance with the requirements of the AES. unlike Serpent and Twofish, the key size must be one of these values; it is not allowed to be arbitrary,**
4. **Although the official AES block size is 128 bits. Both block length and key length can be extended very easily to multiples of 32 bits.**
5. **The number of rounds, of the main algorithm, can vary from 10 to 14 and is dependent on the block size and key length.**

6. **The low number of rounds has been one of the main drawbacks of Rijndael, but if this ever becomes a problem the number of rounds can easily be increased at little extra cost by increasing the block size and key length.**
7. **A data block to be processed using Rijndael is partitioned into an array of bytes, and each of the cipher operations is byte-oriented.**
8. **Rijndael's round function consists of four layers. In the first layer, an 8x8 S-box is applied to each byte. The second and third layers are linear mixing layers, in which the rows of the array are shifted, and the columns are mixed. In the fourth layer, subkey bytes are XORed into each byte of the array. In the last round, the column mixing is omitted.**

7.2 The characteristics of Twofish Algorithm :

1. **The Twofish block cipher is designed to be highly secure and highly flexible.[7]**
2. **It is well suited for large microprocessors, 8-bit smart card microprocessors, and dedicated hardware.**
3. **No attacks can break the full 16 round version of the algorithm. Attacks have been found against a weaker 5 round Twofish, but the algorithm is very secure when the full 16 rounds are used.**
4. **Twofish is a 128-bit block cipher, meaning that data is encrypted and decrypted in 128-bit chunks. The key length can vary, but for the purposes of the AES it is defined to be either 128, 192, or 256 bits.**
5. **It is a Festal network with 16 rounds The Festal structure is slightly modified using 1-bit rotations.**
6. **The round function acts on 32-bit words with four key dependent 8x8 S-boxes.**

7.3The characteristics of Serpent Algorithm:

1. **Serpent was designed by Ross Anderson, Eli Biham and Lars Knudsen as a candidate for the Advanced Encryption Standard. Serpent is faster than DES and more secure than Triple DES.[8]**
2. **The algorithm uses twice as many rounds as are necessary to block all currently known shortcut attacks. This means that Serpent should be safe against as yet unknown attacks that may be capable of breaking the standard 16 rounds used in many types of encryption**
3. **The round function consists of three layers: the key XOR operation, 32 parallel applications of one of the eight specified 4x4 S-boxes, and a linear transformation.**
4. **In the last round, a second layer of key XOR replaces the linear transformation.**

8-The flowcharts Of AES Algorithms:

Figure (1) below show the general flowchart the AES Algorithms and figure (2) show the Flowchart of Encryption & Decryption functions of AES ,And Figure(3) show the flowchart for time function.

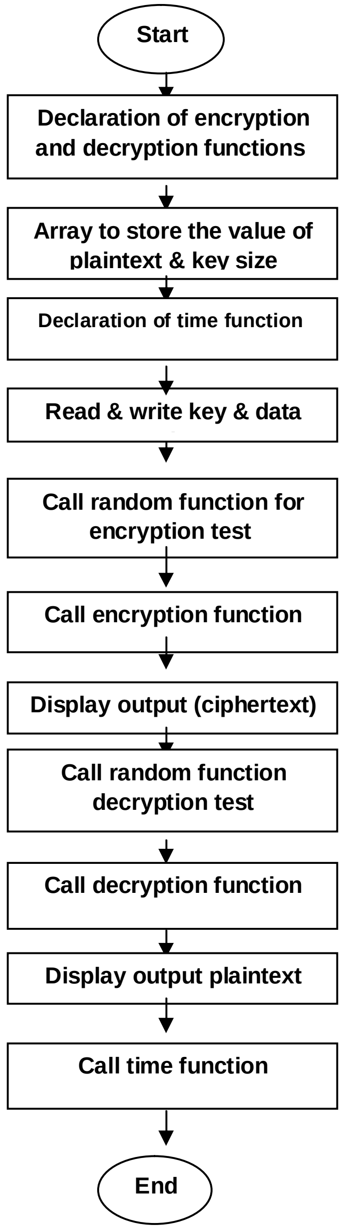
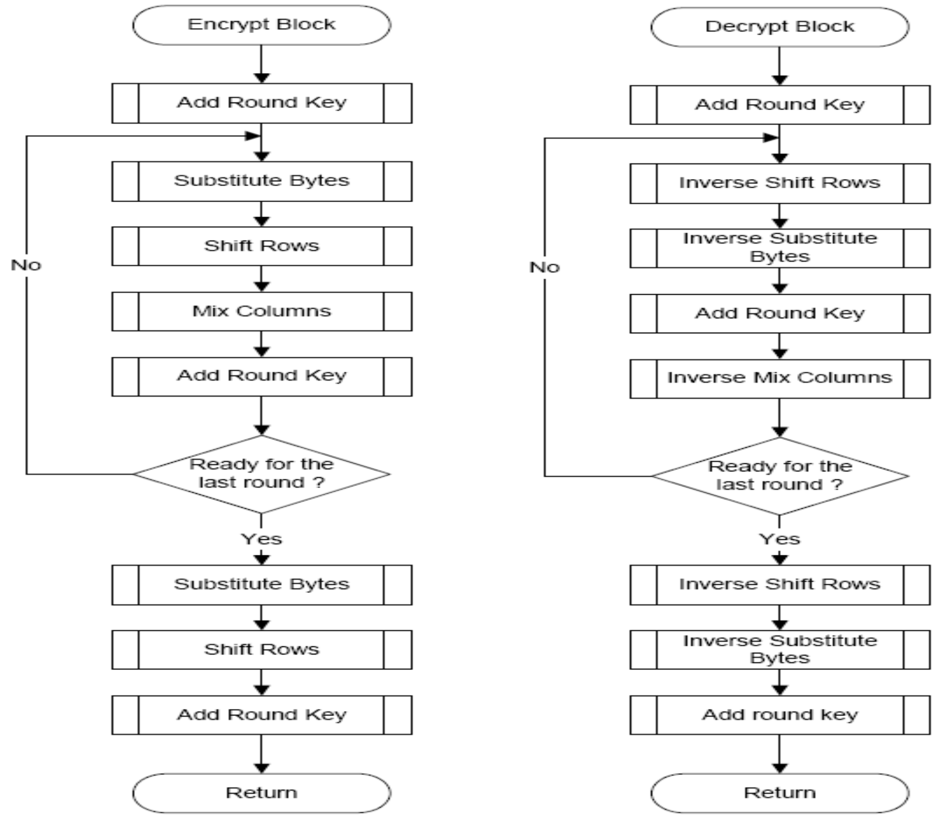

Start
Declaration of encryption and decryption functions
Array to store the value of plaintext & key size
Declaration of time function
Read & write key & data
Call random function for encryption test
Call encryption function
Display output (ciphertext)
Call random function decryption test
Call decryption function
Display output plaintext
Call time function
End

Figure 1 The general Flow Chart of AES



Figure(2) Flowchart of Encryption & Decryption

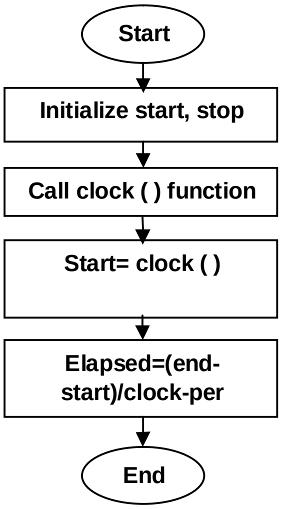
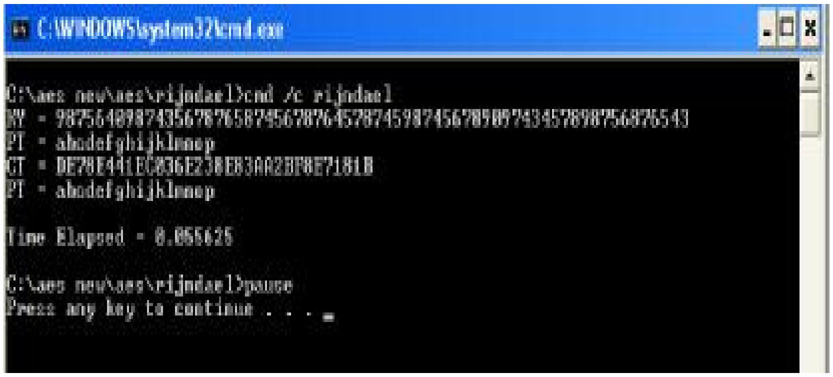


Figure (3) flowchart for time function

8.1 Implementation of the three algorithms: The three algorithms (rijndael ,serpent & twofish) was implemented in two different platforms. They are implemented with three different data for key and plain text on laptop pc.(intel core2 duo 2GB). And three different data for key and plain text on desktop pc (samsung 3GB). And the time required for decryption and encryptionin three algorithms was calculated and we get the following results:-

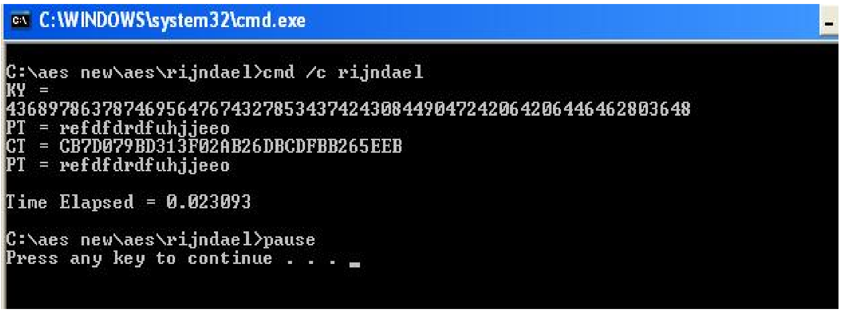
8.2 implementation On Laptop Pc (2GH$_z$ processor & 2GB RAM):

a) Rijndael implementation: Figures (4.1,4.2,4.3)show the implementation of Rijndal On Laptop Pc With 256key And 128 Bits Block Cipher with three different data the program ask the user to enter the key and plain text then the program display the cipher text and the original plain text with the time require for encryption and decryption.



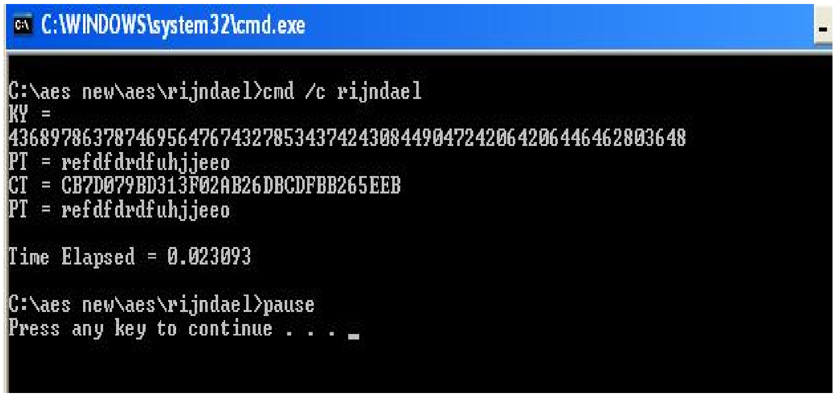
```
C:\aes new\aes\rijndael>cmd /c rijndael
KY = 9875640987435678765874567876457874598745678909743457898756876543
PT = abcdefghijklmnop
CT = DE78E441EC036E238EB3AA2BF8E7181B
PT = abcdefghijklmnop

Time Elapsed = 0.055625

C:\aes new\aes\rijndael>pause
Press any key to continue . . .
```

Figure (4.1) Rijndael Implementation on Laptop pc with 256bits key size &128bits block



```
C:\aes new\aes\rijndael>cmd /c rijndael
KY =
4368978637874695647674327853437424308449047242064206446462803648
PT = refdfdrdfuhjjeeo
CT = CB7D079BD313F02AB26DBCDFBB265EEB
PT = refdfdrdfuhjjeeo

Time Elapsed = 0.023093

C:\aes new\aes\rijndael>pause
Press any key to continue . . .
```

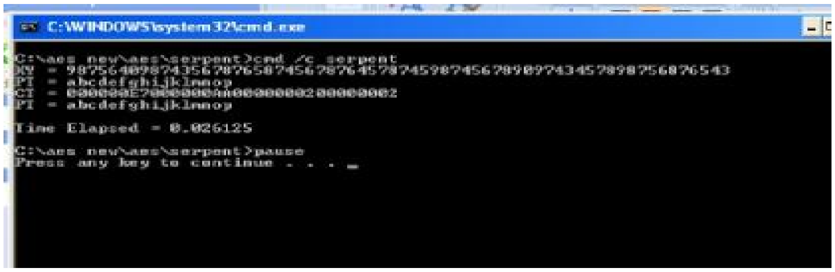
Figure (4.2) Rijndael Implementation on Laptop pc with different key and block



```
C:\aes new\aes\rijndael>cmd /c rijndael
KY =
4368978637874695647674327853437424308449047242064206446462803648
PT = refdfdrdfuhjjeeo
CT = CB7D079BD313F02AB26DBCDFBB265EEB
PT = refdfdrdfuhjjeeo

Time Elapsed = 0.023093

C:\aes new\aes\rijndael>pause
Press any key to continue . . . _
```

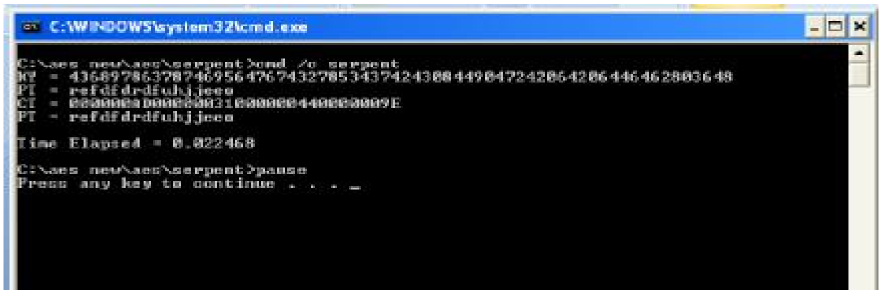
Figure (4.3) Rijndael Implementation On Laptop pc with different key and block

b) Serpent Implementation:Figures(5.1,5.2,5.3) show the implementation of Serpent algorithms On Laptop Pc With 256key And 128 Bits Block Cipher with three different data the program ask the user to enter the key and plain text then the program display the cipher text and the original plain text with the time require for encryption and decryption.



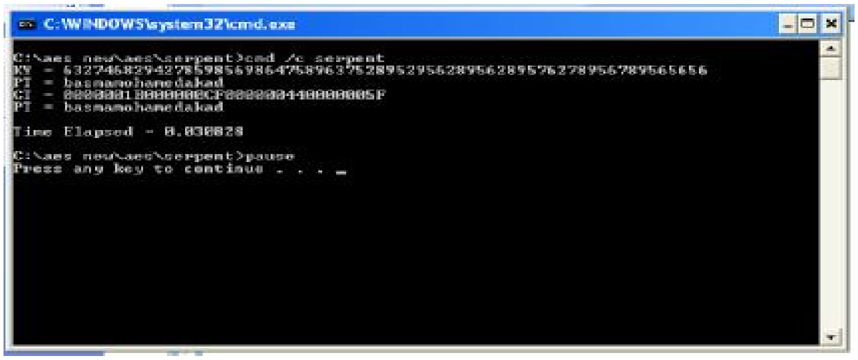
```
C:\aes new\aes\serpent>cmd /c serpent
KY = 9875640987435678765874567876457874598745678909743457898756876543
PT = abcdefghijklmnop
CT = 000000E70000000A0000000200000002
PT = abcdefghijklmnop

Time Elapsed = 0.026125

C:\aes new\aes\serpent>pause
Press any key to continue . . . _
```

Figure (5.1) serpent Implementation on Laptop pc with 256bits key size &128bits block



```
C:\aes new\aes\serpent>cmd /c serpent
KY = 4368978637874695647674327853437424308449047242064206446462803648
PT = refdfdrdfuhjjeeo
CT = 0000000D00000031000000440000009E
PT = refdfdrdfuhjjeeo

Time Elapsed = 0.022468

C:\aes new\aes\serpent>pause
Press any key to continue . . . _
```

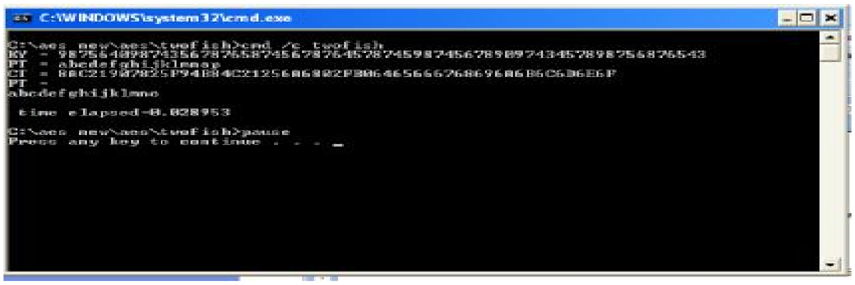
Figure (5.2) Serpent Implementation on Laptop pc with 256bits key size &128bits block



```
C:\aes new\aes\serpent>cmd /c serpent
KY = 63274682942785985698647589637528952955628956289576278956789565656
PT = basmamohamedakad
CT = 00000001B000000CF000000440000000005F
PT = basmamohamedakad

Time Elapsed = 0.030028

C:\aes new\aes\serpent>pause
Press any key to continue . . . _
```

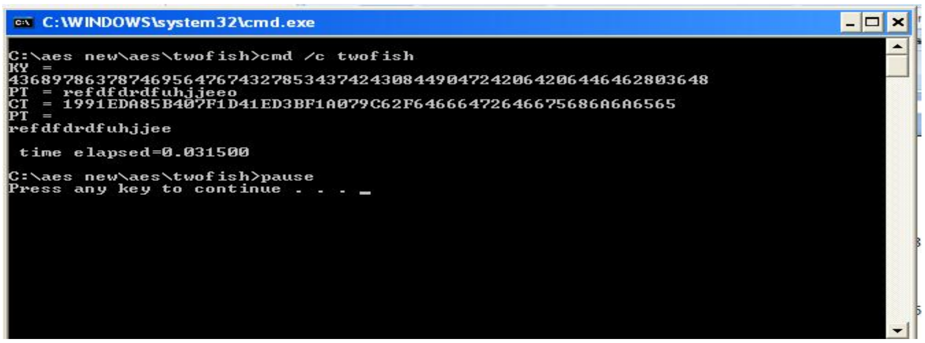
Figure (5.3) Serpent Implementation on Laptop pc with 256bits key size &128bits block

c)Twofish implementation: Figures(6.1,6.2,6.3) show the implementation of twofish algorithms On Laptop Pc With 256key And 128 Bits Block Cipher with three different data the program ask the user to enter the key and plain text then the program display the cipher text and the original plain text with the time require for encryption and decryption.



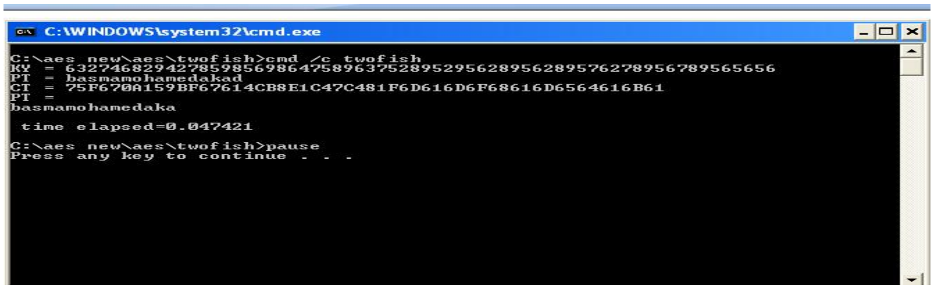
```
C:\aes new\aes\twofish>cmd /c twofish
KY = 9875640987435678765874567876457874598745678909743457898756876543
PT = abcdefghijklmnop
CT = 88C219087825F94B84C2125686802F806465666768696A6B6C6D6E6F
PT =
abcdefghijklmno

 time elapsed=0.028953

C:\aes new\aes\twofish>pause
Press any key to continue . . . _
```

Figure (6.1) Towfish Implementation on Laptop pc with 256bits key size &128bits block



```
C:\aes new\aes\twofish>cmd /c twofish
KY =
4368978637874695647674327853437424308449047242064206446462803648
PT = refdfdrdfuhjjeeo
CT = 1991EDA85B407F1D41ED3BF1A079C62F6466647264667568 6A6A6565
PT =
refdfdrdfuhjjee

 time elapsed=0.031500

C:\aes new\aes\twofish>pause
Press any key to continue . . . _
```

Figure (6.2) Towfish Implementation on Laptop pc with 256bits key size &128bits block

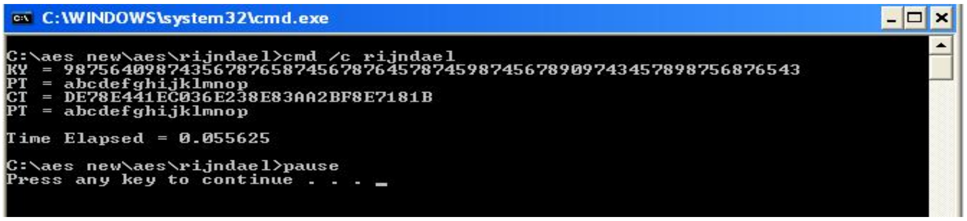


```
C:\aes new\aes\twofish>cmd /c twofish
KY  = 6327468294278598569864758963752895295628956289576278956789565656
PT  = basmamohamedakad
CT  = 75F670A159BF67614CB8E1C47C481F6D616D6F68616D6564616B61
PT  =
basmamohamedaka

 time elapsed=0.047421

C:\aes new\aes\twofish>pause
Press any key to continue . . .
```

Figure (6.3) Towfish Implementation On Laptop pc with 256bits key size &128bits block

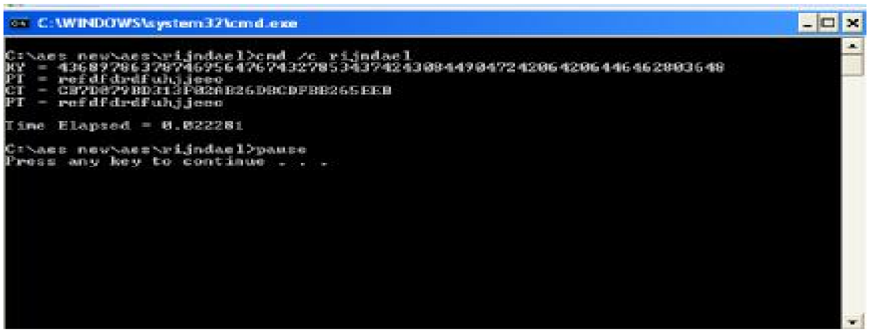
8.2.1 Implementation on desktop pcsamsung (3GH$_Z$ processor & 3GB RAM):

a)Rijndael implementation:figures (7.1,7.2,7.3)show the implementation of rijndal On desktop pc..samsung (3GH$_Z$ processor & 3GB RAM)With 256key And 128 Bits Block Cipher with three different data the program ask the user to enter the key and plain text then the program display the cipher text and the original plain text with the time require for encryption and decryption.



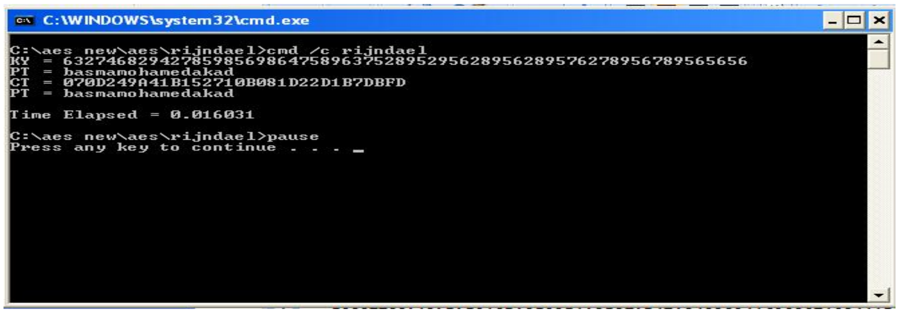
```
C:\aes new\aes\rijndael>cmd /c rijndael
KY  = 9875640987435678765874567876457874598745678909743457898756876543
PT  = abcdefghijklmnop
CT  = DE78E441EC036E238E83AA2BF8E7181B
PT  = abcdefghijklmnop

Time Elapsed = 0.055625

C:\aes new\aes\rijndael>pause
Press any key to continue . . . _
```

Figure (7.1) Rijndael Implementation on desktop pc with 256bits key size &128bits block



```
C:\aes new\aes\rijndael>cmd /c rijndael
KY  = 4368978637874695647674327853437424308447047242064206446462803648
PT  = refdfdrdfuhjjeee
CT  = CB7D079BD313F02AB26D8CDFB8265EEB
PT  = refdfdrdfuhjjeee

Time Elapsed = 0.022281

C:\aes new\aes\rijndael>pause
Press any key to continue . . .
```

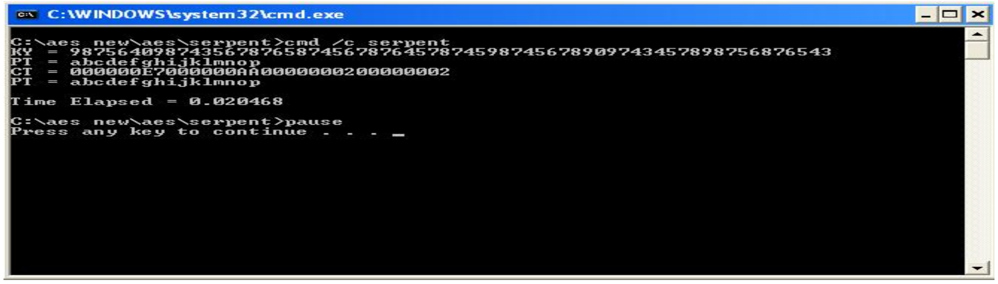
Figure (7.2) Rijndael Implementation on desktop pc with 256bits key size &128bits block



```
C:\aes new\aes\rijndael>cmd /c rijndael
KY  = 6327468294278598569864758963752895295628956289576278956789565656
PT  = basmamohamedakad
CT  = 070D249A41B152710B081D22D1B7DBFD
PT  = basmamohamedakad

Time Elapsed = 0.016031

C:\aes new\aes\rijndael>pause
Press any key to continue . . . _
```

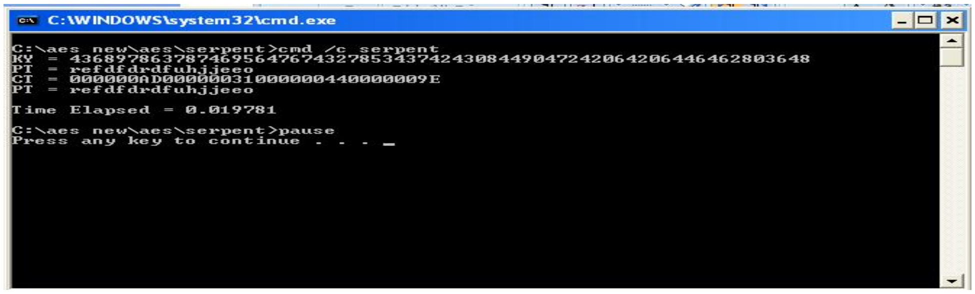
Figure (7.3) Rijndael Implementation on desktop pc with 256bits key size &128bits block

Serpent implementation figures (8.1,8.2,8.3)show the implementation of serpent On desktop pc..samsung (3GH$_Z$ processor & 3GB RAM)With 256key And 128 Bits Block Cipher with three different data the program ask the user to enter the key and plain text then the program display the cipher text and the original plain text with the time require for encryption and decryption.



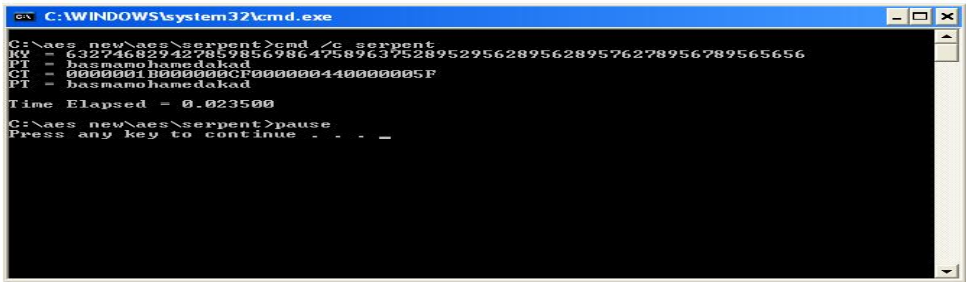
```
C:\aes new\aes\serpent>cmd /c serpent
KY  = 9875640987435678765874567876457874598745678909743457898756876543
PT  = abcdefghijklmnop
CT  = 000000E7000000AA00000000200000002
PT  = abcdefghijklmnop

Time Elapsed = 0.020468

C:\aes new\aes\serpent>pause
Press any key to continue . . . _
```

Figure (8.1) Serpent Implementation on desktop pc with 256bits key size &128bits block



```
C:\aes new\aes\serpent>cmd /c serpent
KY  = 4368978637874695647674327853437424308449047242064206446462803648
PT  = refdfdrdfuhjjeeo
CT  = 000000AD00000003100000000440000009E
PT  = refdfdrdfuhjjeeo

Time Elapsed = 0.019781

C:\aes new\aes\serpent>pause
Press any key to continue . . . _
```

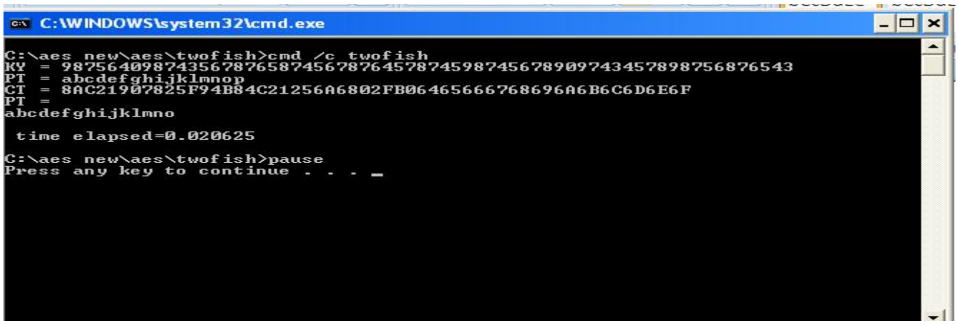
Figure (8.2) Serpent Implementation on desktop pc with 256bits key size &128bits block



```
C:\aes new\aes\serpent>cmd /c serpent
KY  = 6327468294278598569864758963752895295628956289576278956789565656
PT  = basmamohamedakad
CT  = 0000001B000000CF00000044000005F
PT  = basmamohamedakad

Time Elapsed = 0.023500

C:\aes new\aes\serpent>pause
Press any key to continue . . . _
```

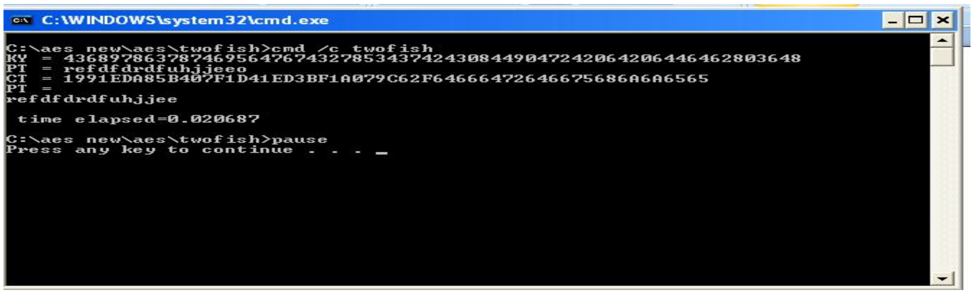
Figure (8.3) Serpent Implementation on desktop pc with 256bits key size &128bits block

b)Twofish implementation :figures (9.1,9.2,9.3)show the implementation of twofish on desktop pc..samsung (3GH$_Z$ processor & 3GB RAM)With 256key And 128 Bits Block Cipher with three different data the program ask the user to enter the key and plain text then the program display the cipher text and the original plain text with the time require for encryption and decryption.



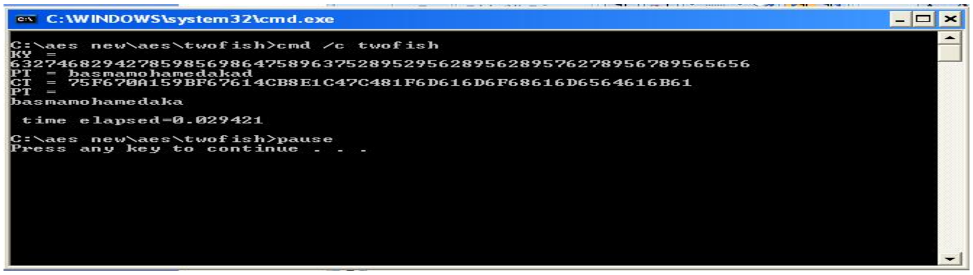
```
C:\aes new\aes\twofish>cmd /c twofish
KY  = 98756409874356787658745678764578745987456789097434578987568765 43
PT  = abcdefghijklmnop
CT  = 8AC21907825F94B84C21256A6802FB0646566676869 6A6B6C6D6E6F
PT  =
abcdefghijklmno

 time elapsed=0.020625

C:\aes new\aes\twofish>pause
Press any key to continue . . . _
```

Figure (9.1) Towfish Implementation on desktop pc with 256bits key size &128bits block



```
C:\aes new\aes\twofish>cmd /c twofish
KY  = 4368978637874695647674327853437424308449047242064206446462803648
PT  = refdfdrdfuhjjeeo
CT  = 1991EDA85B407F1D41ED3BF1A079C62F646664726466756 86A6A6565
PT  =
refdfdrdfuhjjee

 time elapsed=0.020687

C:\aes new\aes\twofish>pause
Press any key to continue . . . _
```

Figure (9.2) Towfish Implementation on desktop pc with 256bits key size &128bits block

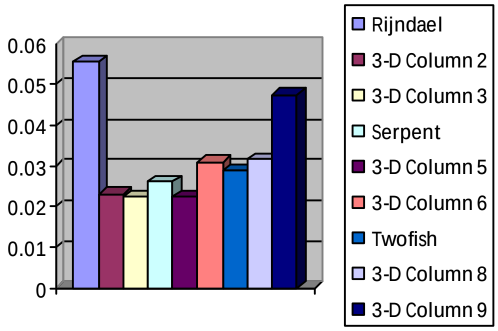


```
C:\aes new\aes\twofish>cmd /c twofish
KY  =
6327468294278598569864758963752895295628956289576278956789565656
PT  = basmamohamedakad
CT  = 75F670A159BF67614CB8E1C47C481F6D616D6F68616D6564616B61
PT  =
basmamohamedaka

 time elapsed=0.029421

C:\aes new\aes\twofish>pause
Press any key to continue . . .
```

Figure (9.3) Towfish Implementation on desktop pc with 256bits key size &128bits block

Table 1 the result of implementation of three algorithms on Laptop pc

Algorithm-name	Time elapsed on Laptoppc.(Dell)	RAM	Processor
Rijndael	0.055625	2GB	2GH$_Z$
	0.023093	2GB	2GH$_Z$
	0.022359	2GB	2GH$_Z$
Serpent	0.026125	2GB	2GH$_Z$
	0.022468	2GB	2GH$_Z$
	0.030828	2GB	2GH$_Z$
Twofish	0.028953	2GB	2GH$_Z$
	0.031500	2GB	2GH$_Z$
	0.047421	2GB	2GH$_Z$



Figure(10) Histogram of the result of table1

Table 2 the result of implementation of three algorithms on Desktop pc

Algorithm-name	Time elapsed on Desktoppc.(Samsung)	RAM	Processor
Rijndael	0.055625	3GB	3GH$_Z$
	0.023093	3GB	3GH$_Z$
	0.022359	3GB	3GH$_Z$
Serpent	0.026125	3GB	3GH$_Z$
	0.022468	3GB	3GH$_Z$
	0.030828	3GB	3GH$_Z$
Twofish	0.028953	3GB	3GH$_Z$
	0.031500	3GB	3GH$_Z$
	0.047421	3GB	3GH$_Z$

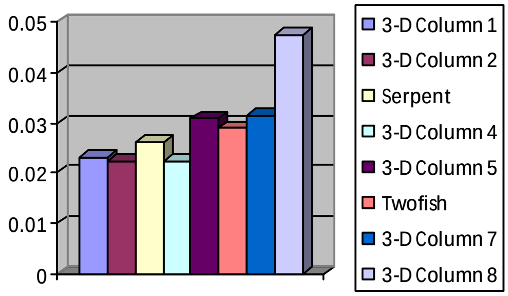


Figure (11) Histogram of the result on table2

Conclusion:

From our implementation of the three algorithms using assembly languages on different platforms we can conclude the following:

1-When the computer is laptop pc with the processor intel 2GH and the RAM 2GB and block size is 128bits and key size is 256bits the best efficient algorithm are as in descending order Rijendal, Serpent,then Towfish.

2-When the computer is Desktoppc.(Samsung)with the processor intel 3GH and the RAM 3GB and block size is 128bits and key size is 256bits the best efficient algorithm are as in descending order Serpent ,Rijendal,then Towfish.

3- Rijndael's key length is defined to be either 128, 192, or 256 bits unlike Serpent and Twofish, the key size must be one of these values; it is not allowed to be arbitrary,

References

[1]James Nechvatal,Elaine Barker, Lawrence Bassham, William Burr,Morris Dworkin, Jame Foti, Edwar Roback Report on the Development of the Advanced Encryption Standard Technology Laboratory National Institute of Standards and Technology Publication Date: October 2, 2000.

[2]William stallings,Cryptography and network security,Principles and Practices, Fourth Edition,November 16,2005,prentice Hall.

[3]FIPS, "Advanced Encryption Standard." National Institute of Standards and Technology Nov.2001.

[4] J. Worley, et al., AES Finalists on PA-RISC and IA-64: Implementations &Performance, in The Third AES Candidate Conference, printed by the National Institute of Standards and Technology, Gaithersburg, MD, April 13-14, 2000, pp. 57-74.

[5]F. Sano, et al., Performance Evaluation of AES Finalists on the High-End Smart Card, in The Third AES Candidate Conference, printed by the National Institute of Standards and technology, Gaithersburg, MD, April 13-14, 2000, pp. 82-93.

[6]J. Daemon and V. Rijmen, AES Proposal: Rijndael, AES algorithm submission, Sep 1999

[7]B. Schneier, et al., Twofish: A 128-Bit Block Cipher, AES algorithm submission ,Jun15,1998

[8]R. Anderson, E. Biham, and L. Knudsen, Serpent: A Proposal for the Advanced Encryption Standard, AES algorithm submission, June 1998.

[9]Bart Van Rompay. Analysis and design of cryptographic hash functions ,Mac algorithms and block ciphers,2004 Phd thesis, katholicke University Leuven ,B.preneel and J.vandewalle(promoters),240:10-14.

قياس كفاءة الاداء للقائمة الاخيرة المرشحة لخوارزميات التشفير المتقدمة

م. هناء رشيد اسماعيل

جامعة النهرين

المستخلص:

ان نظام الشفرة المعيارية المتقدمة هو من الانظمة المستخدمة للخوارزميات المتناظرة المفتاح.لقد اختار المعهد الامريكي للمعايير و التكنولوجية وبعد عدة بحوث و دراسات افضل خمس خوارزميات متناظرة المفتاح وهي .
MARS , RC6 ,Rijndael , Serpent and Towfish

يهدف البحث الى قياس كفاءة الاداء لثلاث خوارزميات وهي Rijndael , Serpent and Towfish

ان سبب اختيارنا لهذه الخوارزميات هو اشتراكهم بكتلة ثابتة الحجم هي 128 بت وحجم مفتاح (128و192و256) بت .

تم قياس سرعة الاداء للخوارزميات الثلاثة وتم حساب الوقت لكل خوارزمية في التجفير وفي استرجاع النص تم تنفيذ البرامج على الحاسبات :

Desktop pc.(samsong 3GB) And Laptop pc.(intel core2 2GB)

تم مقارنة النتائج وتبين لنا ان Rijndael هي الاكثر كفاءة في حالة التنفيذ على الحاسبة Laptop pc.(intel core2 2GB) وان Serpent هي الاكثر كفاءة عند التنفيذ على الحاسبة Desktop pc.(samsong 3GB).