Apply Block Ciphers Using Tiny Encryption Algorithm (TEA)

Hana Rashied Ismaeel*

Date of acceptance 19/5 / 2009

Abstract:

Ciphers can be either symmetric-key or public-key. Symmetric-key ciphers are the most famous and important elements in many cryptographic systems. Individually, they provide confidentiality and privacy. The Aim of this paper is to Apply symmetric block ciphers algorithm which is called Tiny Encryption algorithm (TEA). There are many types of block cipher use different techniques and functions using basic arithmetic operations such as XOR, addition, subtraction, multiplication, bitwise shift, and division. TEA cipher uses XOR, addition and bitwise shift. The paper presents how a block cipher could be constructed in general, it includes an observation of the history, inventors, and algorithms of the TEA block ciphers. It also discuss the manner did the programming took in terms of modularity, simplicity, resource reservation, in the previous algorithm the user cannot enter the plain text as a characters, it must enter the ASCII cod of the characters we write a lot e of functions to solve this problem also we enable the user to save his message in text file to make him use it to receive a message from email ,Also the time of encryption and decryption are calculated to measure the performance of the algorithm., The implementation of TEA is made using C/C++ Borland compiler version(4.5).

Key words: TEA , Symmetric-key , Encryption , Decryption ,Cryptography , Feistel Ciphers, plaintext , cipher text , block ciphers

Introduction:

As computer systems become more persistent and complex, security is increasingly important. Cryptographic algorithms and protocols consists the central component of systems that protect network transmissions and store data. The security of such systems greatly depends on the methods used to manage, establish, and distribute the keys employed by the cryptographic techniques. Even if a cryptographic algorithm is ideal in both theory and Implementation, the strength of the algorithm will be rendered useless if the relevant keys are poorly managed. Cryptography is the art and science behind the principles, means, and methods for keeping messages secure. Cryptanalysis is a study of how to compromise cryptographic mechanism [1]. There are two classes of key-based encryption algorithms: symmetric (or secret-key) and asymmetric (or public-key) algorithms. Symmetric algorithms use the same key for encryption and decryption, whereas asymmetric algorithms use different keys for encryption and decryption. Ideally it is infeasible to compute the decryption key from the encryption key. The symmetric key provides privacy and confidentiality of transferring data in a secure manner via a network or by message transferring. One type of key-based encryption is the block cipher which transforms a constant block of plaintext into a cipher text depending on an algorithm which mainly on the encryption key. There are many types

*College Of Information engineering/ Al Nahrain University

of block cipher each use different techniques and functions using basic arithmetic operations such as XOR, addition, subtraction, multiplication, bitwise shift, and division. The research present TEA cipher that use XOR, addition and bitwise shift.

1.Introduction to Block Ciphers:

The main focus of this paper is symmetric-key block ciphers. A block cipher is a function which maps n-bit plaintext blocks to n-bit cipher text blocks; n is called the block length. It may be viewed as a simple substitution cipher with large character size. The function is parameterized by a k-bit key K, taking values from a subset K (the key space) of the set of all k-bit vectors V_k. It is generally assumed that the key is chosen at random. Use of plaintext and cipher text blocks of equal size avoids data expansion. To allow unique decryption, the encryption function must be one-to-one (i.e., invertible).For n-bit plaintext and cipher text blocks and a fixed key, the encryption function is a bijection, defining a permutation on n-bit vectors. Each key potentially defines a different bijection. The number of keys is |K|, and the effective key size is $\log_2$ |K|; this equals the key length if all k-bit vectors are valid keys ($K = V_k$). **[1].**

1.1 Definition of block cipher: An n-bit block cipher is a function E: Vn × K → Vn, such that for each key k Є K, E (P, K) is an invertible mapping (the encryption function for K) from Vn to Vn, written EK (P). The inverse mapping is the decryption function, denoted P=DK(c). C = EK (P) denotes that cipher text C results from encrypting plaintext P under K **,**Block ciphers process plaintexts in blocks of a fixed length of b bits, typical lengths are b = 64 or b = 128 bits **[2].**

1.2 Modes of operation: Many applications of block ciphers require the encryption of plaintext strings of variable length. The most straight forward approach to realize this with a b-bit block cipher is to divide the plaintext in separate blocks of b bits and encrypt these blocks independently. The cipher text consists of the concatenation of the encrypted blocks. This procedure is called the Electronic Code Book (ECB) mode of operation. Note that a well-defined padding rule must be used to pre-process the plaintext in such a way that the input length is guaranteed to be a multiple of b bits. A disadvantage of the ECB mode is that repetitions of blocks in the plaintext lead to equal cipher text blocks. This problem is solved in the Cipher Block Chaining (CBC) mode, where the value of a cipher text block depends not only on the corresponding plaintext block but also on the previous cipher text block. **[3].**

1.3 Design of Block Ciphers: In practice, block ciphers are designed in a manner similar to the one used for cryptographic hash functions **[4]**: designers learn from mistakes made in the past and try to prevent known methods of attack. There are no block ciphers which are both practical and provably secure, although in some cases bounds can be given for the complexity of specific known attacks. Confidence in a new design is only obtained when it does not get broken after a substantial amount of expert cryptanalysis **[5]**.

1.3.1 Choice of the parameters: There are two important parameters in the design of a block cipher: the length of the key (k bits) and the length of the blocks (b bits). The key length should be chosen large enough to make exhaustive key search infeasible. In choosing the block length one has to take into account the so-called matching cipher text attack. Because of only about $2^{b/2}$ cipher text blocks are needed to obtain a matching pair, **[6]** and this leaks information on the plaintext, therefore it is recommended that a single key is used to encrypt at most $2^{b/2}$ blocks. Most designs have a block length of b = 64 bits (e.g., DES) or b = 128 bits (e.g., AES).

1.3.2 Block Cipher Constructions: Most known block ciphers have an internal iterated structure which consists of a concatenation of identical round transformations. The idea is that the round transformation is easy to describe and implement, and that by repeating sufficient times, a strong encryption function is obtained. A round transformation can be described by X_i = round (K_i, X_{i-1}), where X_{i-1} is the input to the ith round, X_i the output and K_i a round key. An encryption function of r rounds then consists of initializing with the plaintext block X_0 = P, computing the values X_i for i = 1, 2. ,r and finally setting the cipher text block C = X_r. The round keys K_i ($1 \leq i \leq r$) are derived from the encryption key K by means of a separate key scheduling algorithm **[7].**

1.3.3 Components of a Cipher: Typical designs use a round transformation consisting of several distinct components, each with their own functionality. The purpose of non-linear substitution boxes (or S-boxes) is to achieve confusion that is a complex mixing of bits which are close to each other. This is usually implemented by means of table look-ups. Bit permutations can be used to achieve diffusion that is a re-arranging of bits so that bits which are close to each other in one round are not close to each other in the next round **[8].**

1.3.4 Golden ratio: In mathematics and the arts, two quantities are in the golden ratio if the ratio between the sum of those quantities and the larger one is the same as the ratio between the larger one and the smaller. The golden ratio is approximately 1.6180339887.

1.3.5 Feistel Ciphers: Feistel ciphers are a special type of block ciphers, where the intermediate values X_{i-1} are divided into two halves (L_{i-1}, R_{i-1}) and where the round transformation is based on a round function F which depends on the round key K_i and operates on one half of the input. The output of F is added to the other half of the input, and finally the two halves are swapped to obtain the output of the round transformation. This means that the round transformation (L_{i-1}, R_{i-1}) = round (K_i, (L_{i-1}, and R_{i-1})) has the structure shown in equations 1 and 2:

$$R_i = L_{i-1} \oplus F(K_i, R_{i-1}) \quad ...1$$
$$L_i = R_{i-1}. \quad ...2$$

In the last round the swapping of the two halves is omitted. The advantage of the Feistel construction is that the decryption operation is the same as the encryption operation, except that the round keys need to be used in reverse order **[9].** This minimizes the implementation cost when both encryption and decryption are needed.

2-Tiny Encryption Algorithm The Tiny Encryption Algorithm is a Feistel type cipher, that uses operations from mixed (orthogonal) algebraic groups. Figure (1) **is a** flowchart shows the structure of the TEA the plain text is read from text file then call function (text2hex) which call function (key2hex) which convert key to hexadecimal then call tea-encode function to convert the hexadecimal into string finally the decode function is called.

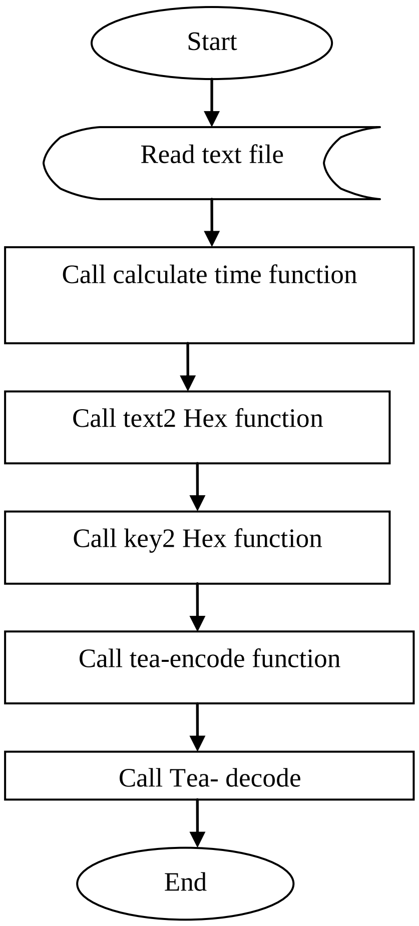


Figure (1) flowchart of TEA

2.1Key2hex: This function takes the key from file then converts it to hexadecimal as shown in figure (2) below.

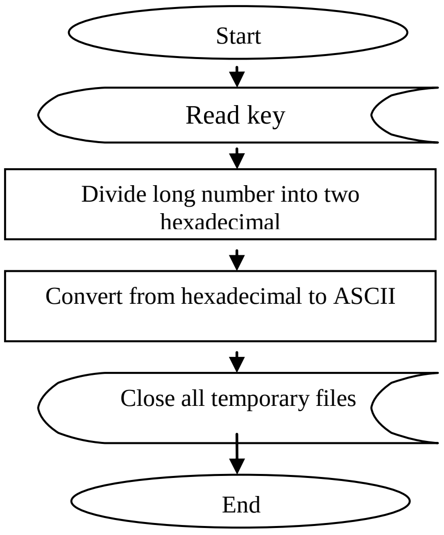

Start
Read key
Divide long number into two hexadecimal
Convert from hexadecimal to ASCII
Close all temporary files
End

Figure 2 key2hex function

2.2Tex2hex: This function takes the plaintext from file as an ASCII format then each letter is converted into two hex-decimal digits (8 bits) as in Figure (3).

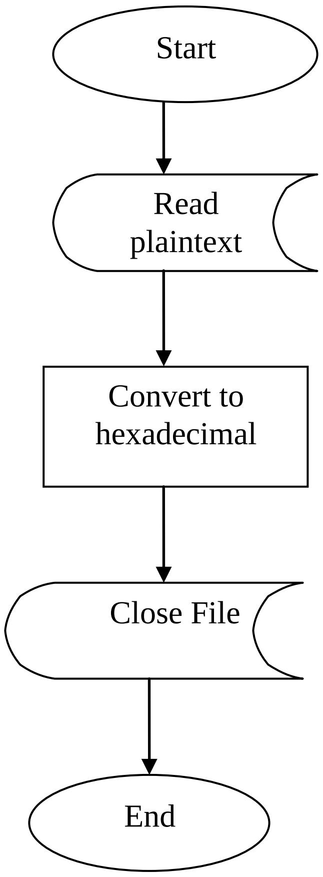


Figure (3) Text 2 Hexadecimal

2.3 Encryption: Figure (4) **is a** flowchart shows the structure of the encryption. The inputs to the encryption function are plaintext block and a key.

The constant delta1 $=(\sqrt{5}-1)*2^{31} = 9E3779B9_h$ is derived from the golden Number ratio to ensure that the sub keys are distinct and its precise value has no cryptographic significance right shift operation.**[10]**

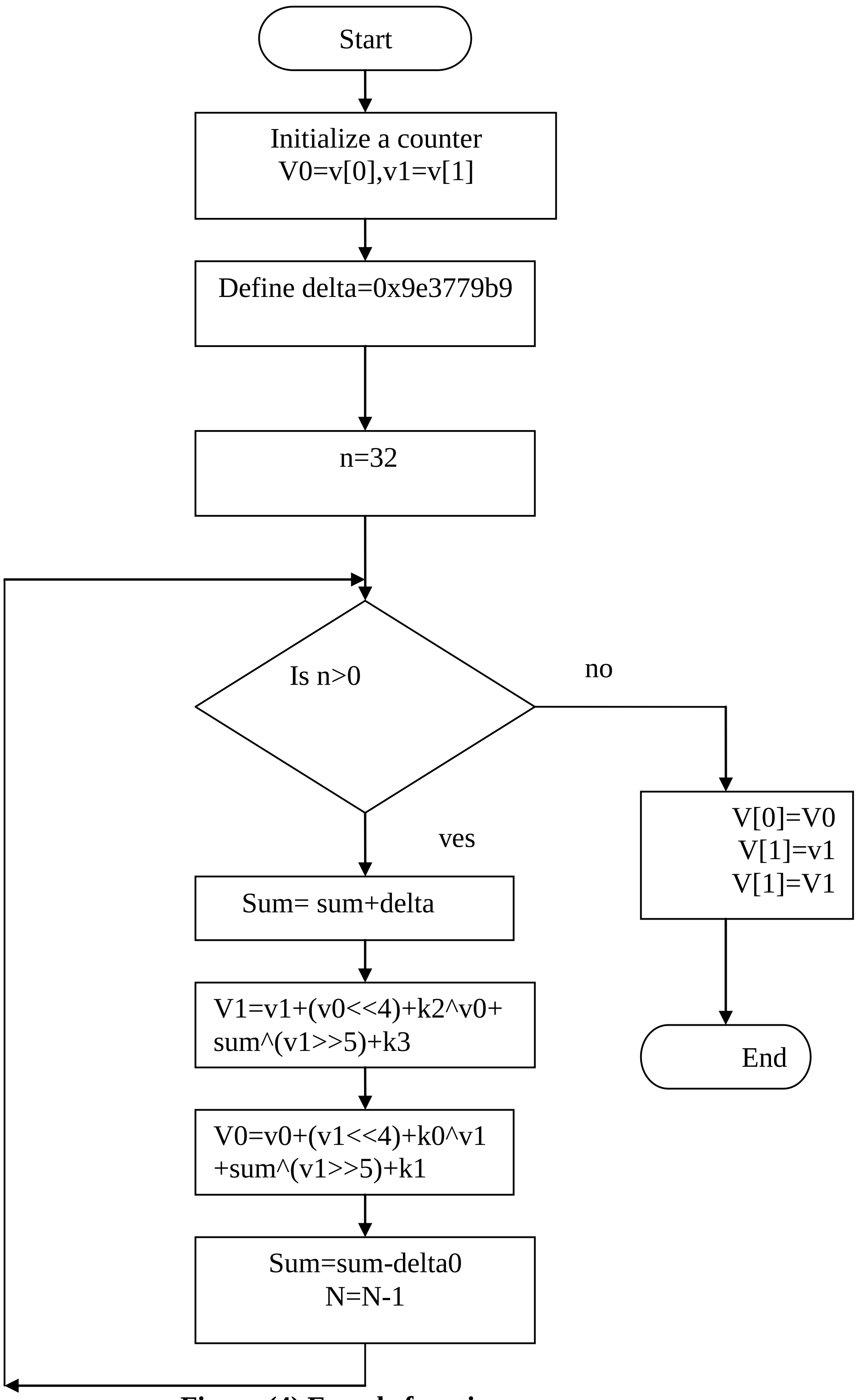


Figure (4) Encode function

2.4 Decryption:

Decryption is essentially the same as the encryption process; in the decode routine the cipher text is used as input to the algorithm, but the sub keys K[i] are used in the reverse order. **Figure (5)** presents the structure of the TEA decryption flowchart.

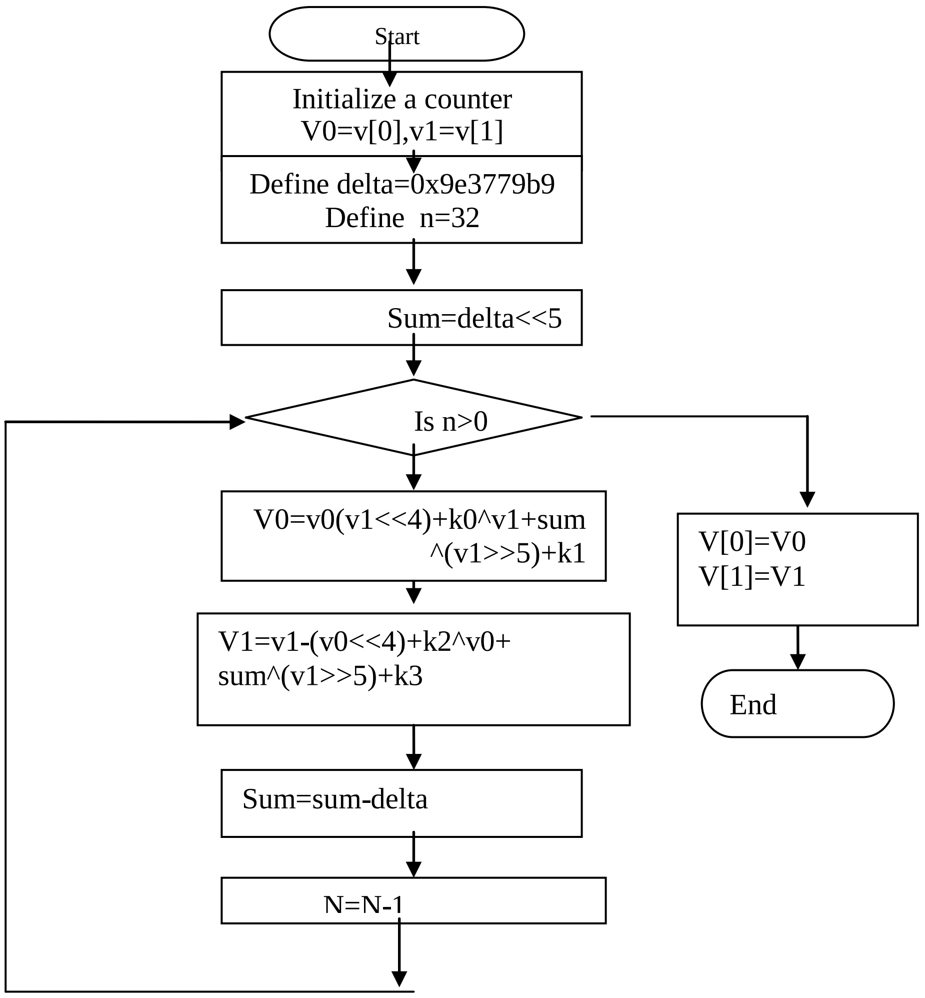


Figure (5) Decode function

2.5clac-time function: This function calculate the elapsed time of the tea algorithm by calling clock (), which is found in time.h header file which is found in c++ library as shown in figure (6) below.

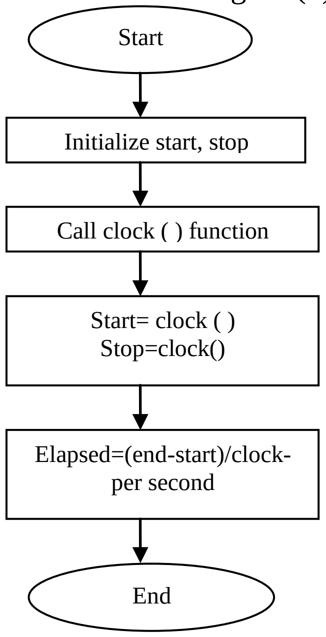
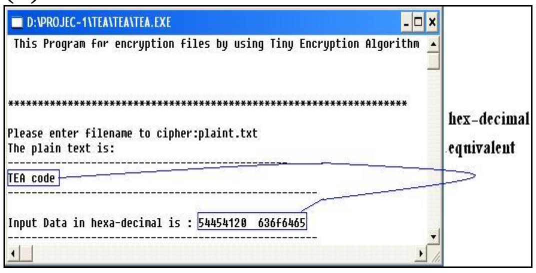


Figure (6) calculate time function

Implementation of Tiny Encryption Algorithm (TEA):

TEA is programmed by using C\C++ programming language. The program takes the 64-bit plain text input in array **v** and 128-bit key in array **k** and gives the cipher text as output on execution. Many steps take to perform this program:

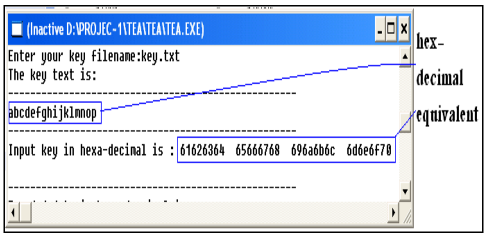
1-Converting the plaintext to hex-decimal notation: Because of plaintext are generally use ASCII format, there must be a mechanism to change it into something that is meaningful by the computer. So it must be converted into a number to perform the ciphering operations such as addition, subtraction, bitwise shift, and so on. The program first take the plaintext from file as an ASCII format then each letter is converted into two hex-decimal digits (8 bits) as in Figure (7).



```
D:\PROJEC~1\TEA\TEA\TEA.EXE
 This Program for encryption files by using Tiny Encryption Algorithm



*******************************************************************

Please enter filename to cipher:plaint.txt
The plain text is:
----------------------------------------------------------
TEA code
----------------------------------------------------------

Input Data in hexa-decimal is : 54454120  636f6465
----------------------------------------------------------
```

Figure (7) Converting plaintext to hex-decimal

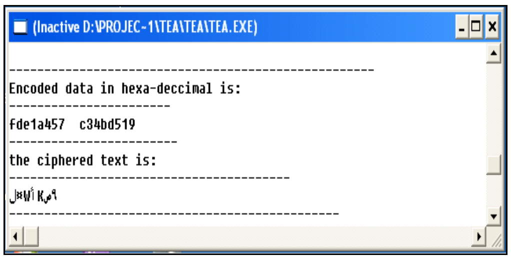
2-Converting the key from text to hex-decimal notation: Usually the key is used as hex-decimal form, but to make the key easy to keep it in mind it will use as text, (but there is one drawback the key will not make full use of 128-bit strength because not all hex-decimal digits represents a symbol that is used in English language).This operation is illustrated below in Figure (8).



```
(Inactive D:\PROJEC~1\TEA\TEA\TEA.EXE)
Enter your key filename:key.txt
The key text is:
----------------------------------------------------------
abcdefghijklmnop
----------------------------------------------------------
Input key in hexa-decimal is : 61626364  65666768  696a6b6c  6d6e6f70


----------------------------------------------------------
```

Figure (8) converting key to hex-decimal

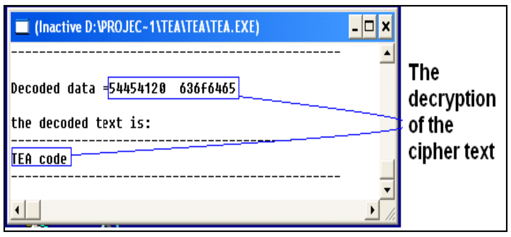
3-Encryption: After the plaintext and the key had been converted to numbers, the encryption routine is called to perform the encryption process according to the algorithm in figure (1) the encryption process is demonstrated in Figure (9).



```
(Inactive D:\PROJEC~1\TEA\TEA\TEA.EXE)

----------------------------------------------------------
Encoded data in hexa-deccimal is:
-----------------------
fde1a457  c34bd519
------------------------
the ciphered text is:
----------------------------------------
ل¤Wأ Kص٩
-----------------------------------------------
```

Figure (9) Encryption

4-Decryption: The decryption routine is called with the parameters of the ciphered text and the same key that was used in the ciphering process. The output of the program is shown below in figure (10):



```
(Inactive D:\PROJEC~1\TEA\TEA\TEA.EXE)
----------------------------------------------------------

Decoded data =54454120  636f6465

the decoded text is:
----------------------------------------------------------
TEA code
----------------------------------------------------------
```

Figure (10) Decryption

Evaluation of TEA: Table (1) below summarize the amount of time elapsed for encryption, decryption and key set-up of (128) block of TEA which is implemented using c++ compiler.

Table (1) time elapsed

Encryption speed(64 bits)	0.248 sec
Decryption speed(64 bits)	0.113 sec
Key set-up (128 bits)	0.163 sec

Discussion:

In the previous algorithm the user cannot enter the plain text as a characters, it must enter the ASCII cod of the characters we wrote a lot e of functions to solve this problem also we enable the user to save his message in text file to be used in different applications such as web services (email message, image, audio,.) Also to make the key easily used it must be converted from hex-decimal to characters as a text, but when we apply the algorithm we find that the key length is not use all (128-bits) because not all hex-decimal digits represents a symbol that is used in English language,. Also the algorithm was evaluated, the time of encryption and decryption is calculated as shown in table (1).

Conclusion:

1-The research is programmed by using C\C++ programming language. The program takes the 64-bits plain text input in array **v** and 128-bit key in array **k** and gives the cipher text as output on execution

2-Because the plain text is generally use ASCII format, it must be converted into a number to perform the ciphering operations such as addition, subtraction, bitwise shift, and so on. The program first take the plaintext from file as an ASCII format then each letter is converted into two hex-decimal digits (8 bits) as in Figure (6).

3-Usually the key is used as hex-decimal form, but to make the key easily used it will use as text, (but there is one drawback the key will not make full use of 128-bit strength because not all hex-decimal digits represents a symbol that is used in English language).This operation is illustrated in Figure (7).

4-The function clock is used to calculate the amount of time elapsed for encryption and decryption of an individual block to measure the performance of the algorithm.

5-The program save the plaintext in text file ,so the algorithm can be used in different applications such as web services (email message, image, audio,..).

References:

1. Alfred J. Menezes, Paul C. van Orschot and Scott A.Vanstone 1997. Handbook of Applied Cryptography.CRC Press Iisbn 0-8493-85237, pp 816 .
2. Andem, Vikram Reddy ,2003. A Cryptanalysis of the Tiny Encryption Algorithm, Masters thesis. Tuscaloosa: The University of Alabama.
3. Wheeler, D.J., & Needham, R.J. 1994. TEA, a tiny encryption algorithm. In Fast Software Encryption – Proceedings of the 2nd International Workshop,1008.
4. FIPS Nov. 2001. specification for the Advanced Encryption Standard. National Institute of Standards and Technology, 197: 14-20
5. John Kelsey, Bruce Schneier, and David Wagner. 1997. Related -Key cryptanalysis of 3-WAY,Biham-DES,CAST,DES-X NEW DES,RC2 and TEA.Lecture Notes in Computer Science,1334: 233–246.
6. Bart Van Rompay. June,2004. Analysis and design of cryptographic hash functions, Mac algorithms and block ciphers,PHD thesis, katholicke University Leuven ,B.preneel and J.vandewalle (promoters), 240:10-14.

7. V. Rijmen. Oct. 1997. Cryptanalysis and Design of Iterated Block Ciphers. Doctoral Dissertation, Katholic University of Leaven.

8. Shannon C. E. Oct, 1994. Communication theory of secrecy systems. Bell System Technical Journal, 28: 656-715.

9. Vikram Reddy Andem. 2003. A Cryptanalysis of the Tiny Encryption Algorithm, Masters Thesis, the University of Alabama, Tuscaloosa.

10. FIPS. Oct, 1999. Data Encryption Standard. National Institute of Standards and Technology. (Specifies the use of Triple-DES), 46-3.

تطبيق خوارزمية التجفير المتناظرة المفتاح (تيَ)

هناء رشيد اسماعيل*

*كلية هندسة المعلومات/جامعة النهرين

الخلاصة:

ان طـرق التجفيـر (Ciphers) تكـون علـى نـوعين امـا متنـاظرة المفتـاح (symmetric-key) او عامـة المفتاح(public-key).وتعتبر الطريقة المتناظرة المفتاح هي الطريقة الاكثر اهمية وشيوعا في عدد كبير من انظمـة التجفير (cryptographic) .ان الهدف من هذا البحث هو تنفيذ الخوارزميةالمتناظرة المفتاح والتي تدعى Tiny Encryption Algorithm(TEA) هذه الطريقة في التجقير تستخدم العمليات الرياضية مثل (addition ,bitwise shift and XOR).تم اضافة دوال الى الخوارزمية. بحيث تمكن المستفيد من ادخال النص و المفتاح بشكل حروف لانه بالخوارزمية الاصلية يتم ادخال النص بال(hexa-decimal)كما سمح للمستخدم باستخدام الملفا ت لخزن النص المراد تجفيره استخدمت لغتي (C and C+ +version 4.5) لتنفيذ البرامج وتم حساب الوقت المستخدم للتجفير لقياس كفاءة الاداء.