A Modification of TEA Block Cipher Algorithm for Data Security (MTEA)

Gaidaa Saeed Mahdi*

Received on: 24/1/2010
Accepted on: 3/3/2011

Abstract

This paper attempts to develop a simple, stronger and safer cryptographic algorithm which would not only be a secure one, but also reduces total time taken for encryption and decryption. The modified algorithm MTEA is a new secret-key block cipher of 64 bit that uses good features of Tiny Encryption Algorithm (TEA) and RC6 algorithms. The proposal algorithm uses the RC6 encryption algorithm as key scheduling to generate subkey. These generated key will be used in META algorithm's round. An effort is made to enhance performance of the resulting algorithm. Proposed MTEA algorithm improved TEA algorithm which is a simple classical Feistel network with 64 rounds and operating on 64 bit blocks of plaintext to produce 64 bit blocks of ciphertext with 128 bit key.

Keywords: Tea, cryptography, Feistel Network, block cipher, AES.

التحديث في التشفير الكتلي TEA لأمنية البيانات (MTEA)

الخلاصة

يحاول هذا البحث تطوير خوارزمية تشفير بسيطة أقوى وأكثر أمانا ، والتي لن تكون فقط أمنة ، ولكن أيضا تقلل الوقت الإجمالي المتخذ لتشفير وفك الشفرة. الخوارزمية المحدثة (MTEA) هي تشفير كتلي جديد ذات مفتاح سري من 64 بت يستخدم ميزات جيدة من خوارزمية التشفير الصغير (TEA) وخوارزمية RC6. الخوارزمية المقترحة تستخدم خوارزمية التشفير RC6 كجدولة لتوليد المفاتيح الفرعية.. وسوف تستخدم هذه المفاتيح في جولة خوارزمية MTEA. الجهد المبذول هو لتحسين الأداء للخوارزمية الناتجة عن ذلك. الخوارزمية المقترحة MTEA تحسن خوارزمية TEA التي هي شبكة كلاسيكية بسيطة من نوع Feistel مع 64 جولة ، وتعمل على كتل 64 بت من النص الصريح لإنتاج كتل 64 بت من النص المشفر مع مفتاح 128 بت.

*Chemical Engineering Department, University of technology/ Baghdad

https://doi.org/10.30684/ etj.29.5.1
University of Technology-Iraq, Baghdad, Iraq/2412-0758

1. Introduction

The security of symmetric cryptosystem is a function of two parameters: the strength of the algorithm and the length of the key. The algorithm must be so secure that there is no better way to break it than with a brute-force attack. The security of the algorithm must be reside in the key, therefore, there is a balance between choosing long key and the time required to complete the enciphering operation [1].

The name of block cipher came from the fact that block cipher encrypts plaintext as blocks. These blocks differ in size between block cipher algorithms, for example, in Data Encryption Standard DES the plaintext is divided into blocks of length 64, but it is 32 in International Data Encryption Algorithm (IDEA). If the length of block cipher equal one then, it will become stream cipher.

The basic ingredients of modern fast software block encryption schemes are computer instructions like ROTATE, ADD, XOR etc. Different subsets of such operations will yield an interesting variety of different permutation groups, e.g. symmetric groups. For example simple pair of ROTATE and an ADDITION module are already powerful enough to generate every possible encryption function on its set of input blocks. On the other hand, any possible combination of ROTATE and XOR operations can only produce a subset of at most $n \times 2^n$ Functions within the symmetric group of order n! [2].

The proposed algorithm MTEA will be attempted to Mix operation from different algebraic group: XOR, addition, rotation and multiplication will be adapted from RC6 encryption algorithm to overcome previous TEA encryption algorithm. Several differences from *TEA* are apparent, including a somewhat more complex key-schedule and a rearrangement of the shifts, XORs and additions [3, 4].

In order to understand the design of proposed algorithm MTEA it is necessary to know about TEA algorithm.

2. The Tiny Encryption Algorithm

In cryptography, the Tiny Encryption Algorithm (TEA) is a block cipher notable for its simplicity of description and implementation (typically a few of code). The cipher was initially presented by (Wheeler and Needham 1994). TEA operates on 64-bit decryption at a time [5], and uses a 128-bit key. It has a Feistel with a suggested 64 rounds, typically implemented in pairs termed cycles. It has an extremely simple key schedule, mixing all of the key material in exactly same way for each cycle. Different multiples of a magic constant are used to prevent simple attacks based on the symmetry of the rounds. The magic constant, 2654435769 or 9E3779B916 is chosen to be 232/φ, where φ is the golden ratio [5]. Figure 1 shows the structure of TEA algorithm.

TEA has a few weaknesses. Most notably, it suffers from equivalent keys — each key is equivalent to three others, which means that the effective key size is only 126 bits [6]. As a result, TEA is especially bad as a cryptographic hash function. This weakness led to a method for hacking Microsoft's Xbox game console, where the cipher was used as a hash function. TEA is also susceptible to a related-key attack which requires 223 chosen plaintexts under a related-key pair, with 232 time complexity [7]. Because of these weaknesses, the proposed algorithm MTEA has been designed.

bit key, then AA, AAA, etc., are equivalent keys.)

3. The RC6 Algorithm

The RC6 algorithm is a block cipher that was one of the finalists in the Advanced Encryption Standard (AES) competition ; the AES competition, sponsored by the National Institute of Standards and Technology (NIST), began in 1997. The RC6 algorithm evolved from its predecessor RC5, a simple and parameterized family of encryption algorithms [8].

RC6, like RC5, consists of three components: a key expansion algorithm, an encryption algorithm, and a decryption algorithm. The parameterization is shown in the following specification: RC6-*w/r/b*, where *w* is the word size, *r* is the non-negative number of rounds, and *b* is the byte size of the encryption key. RC6 makes use of data-dependent rotations. RC6 is based on seven primitive operations as shown in Table 1 [8].

Figure (2) is a pictorial representation of the RC6 encryption process for AES. Figure (3) shows the decryption algorithm for RC6 and figure (4) shows the key schedule with RC6-w/r/b [8].

4. The Proposed MTEA algorithm

MTEA is a symmetric block cipher designed to correct weaknesses in *TEA*. Like *TEA*, *MTEA* is a 64-bit block Feistel network with a 128-bit key and 64 rounds. Figure (5) show the block diagram of an MTEA single round.

In proposed algorithm the magic constants, which was added to the right half of input in TEA, will be eliminated.The proposed algorithm used RC6 algorithm as key generation algorithm to overcome the pervious TEA algorithm weaknesses. The use of subkeys that are generating from RC6 will be diffusion the right half which will be then added to the left half.

4.1 Key schedule

The key schedule is an important component of a block cipher; it computes the round keys from the external key. The subkey generation process is designed to preserve the entire entropy of the key and to distribute that entropy uniformly throughout the subkeys. It is also designed to distribute the set of allowed subkeys randomly throughout the domain of possible subkeys. E8 is 8 rounds of RC6 encryption, used for subkey generation. Let K be an inserted key which is 128 bits. Then is used following recursion: K0=K, K1=E8(K0),...,Ki=E8(Ki-1),...K31=E8(K30). These subkeys K0, K1,…,K31 is used as input for each single round for proposing MTEA encryption algorithm. Each of subkeys is 128 bits i.e K0 (*k[0], k[1], k[2], k[3]).* The underlying philosophy behind RC6 is that simplicity of design yields algorithm that is both easier to implement and using a combination of operation. The Proposed algorithm is designed to adapt RC6 algorithm which is used a full menu of “strong operations” supported in modern computers to achieve better security properties, high speed, and implementation flexibility. RC6 algorithm will be used primitive operations (add, subtract, multiply, exclusive-or, and data-dependent rotate). The subkeys must be precomputed before any data encryption or decryption. The proposed method of subkey calculation requires all subkeys to be calculated advance of any data encryption. In fact, it is impossible to calculate the last subkey without calculating every subkey that comes before. This implementation of precomputing the subkeys is for increased speed.

5. Time Considerations

The time-consuming subkey-generation process for proposed algorithm is eliminated because it required only before any data encryption or decryption. Table 2 and Table 3 show the time required to encrypt and decrypt plaintext in second using TEA algorithm and proposed MTEA algorithm respectively.

6. Security of Proposed MTEA Algorithm

In this section a META algorithm and each function that it performs will be evaluated. Avalanche effect of META is calculated and compared with previous TEA algorithm. Some principles that are useful will be discussed to show that this META algorithm will be adapting the AES requirement by the following:

6.1 Weak and Relative Key:

The key schedule is quite complicated and more importantly has a design that might be viewed as being somewhat incompatible with the structure of the encryption process.

The TEA algorithm suffers from equivalent keys. Each key is equivalent to three and it is also susceptible to a related-key attack. Since the publication of RC6, there have been no reported examples of equivalent or related keys. So that there is no such attacks for the proposing MTEA algorithm.

6.2 Avalanche Effect

In cryptography, the **avalanche effect** refers to a desirable property of cryptographic algorithms, typically block ciphers and cryptographic hash functions. The avalanche effect is evident if, when an input is changed slightly (for example, flipping a single bit) the output changes significantly (e.g., half the output bits flip). In the case of quality block ciphers, such a small change in either the key or the plaintext should cause a drastic change in the ciphertext.

If a block cipher or cryptographic hash function does not exhibit the avalanche effect to a significant degree, then it has poor randomization, and thus a cryptanalyst can make predictions about the input, being given only the output. This may be sufficient to partially or completely break the algorithm. Thus, the avalanche effect is a desirable condition from the point of view of the designer of the cryptographic algorithm or device [9].

This section is prepared for making statistical test on the ciphertext that produced from encryption the plaintext in hexadecimal:

1. 0000000000000000
2. 1111111111111111
3. 6666666666666666
4. 9999999999999999
5. bbbbbbbbbbbbbbbb
6. cccccccccccccccc
7. 5555555566666666
8. 3333333311111111
9. 1111111122222222
10. 4444444433333333

Table (4) shows the avalanche effect on the plaintext when only one bit is changed in the key by using TEA algorithm before improvement.

Table (5) shows the avalanche effect on the same plaintext when only one bit is changed in the same key by using MTEA proposed algorithm.

From Table(4) and Table(5) the average of avalanche effect of TEA algorithm is 30 while the average of the avalanche effect of proposed MTEA algorithm is 37.

Tables 4 and 5 are also show that the changing are 24 to 34 bits and 31 to 43 bits out of 64 bits when performing the algorithm before and after improved respectively which mean that 37.5% to 53.13% of each block of the ciphertext is changed (see figure 6). After performing the proposed MTEA algorithm the changing of each block of the ciphertext is 48.44% to 67.18 %.

7. Conclusions

A secure, compact and simple block cipher algorithm is proposed. It offers good performance a considerable exibility. Furthermore, its simplicity will allow analysts to quickly refine and improve our estimates of its security. It offers much improved security/performance over previous TEA algorithm by taking advantage of the powerful operations supported in today's computers.

The design of proposal algorithm achieved a number of objective. It is used RC6 algorithm to generate subkeys which is efficiently and compactly implemented by software these subkeys are precomputed for faster operation. It will be have no weak keys or related keys. Any weak keys should be explicitly known so they can be weeded out during the key generation process by using RC6. The proposal algorithm uses simple operations that are efficient on microprocessors. All operations should manipulate data in byte-sized blocks. Where possible, operations should manipulate data in 32-bit blocks. It has a good avalanche effect. An average of Avalanche Effect of TEA algorithm is about 30. If the same amount of information is changed in key for MTEA algorithm then the average of Avalanche Effect is about 37 .

References

[1] Bruce Shnier "Applied Cryptography Second Edition Protocols. Algorithms, and Source, and Source Code in C", John Wiley and Sons, Inc., 1996.

[2] Thilo Zieschang, " Combinatorial Properties of Basic Encryption Operations", Advances in Cryptology Eurocrypt'97, International Conference on the Theory And Application of Cryptographic Techniques Konstanz, Germany, May 11-15, 1997 Proceedings, Springer, 1997.

[3] Hong S., Deukjo Hong, Youngdai Ko, Donghoon Chang, Wonil Lee, and Sangjin Lee , "Differential cryptanalysis of TEA and XTEA." In Proceedings of ICISC 2003, 2003.

[4] Ko Y., Seokhie Hong, and Wonil Lee, "Related key differential attacks on 26 rounds of XTEA and full rounds of GOST."In Proceedings of FSE '04, Lecture Notes in Computer Science, Springer-Verlag, 2004.

[5] Hernández, Julio César; Isasi, Pedro; Ribagorda, Arturo ,"An application of genetic algorithms to the cryptoanalysis of one round TEA". Proceedings of the 2002 Symposium on Artificial Intelligence and its Application, 2002. http://www.actapress.com/PDFViewer.aspx?paperId=26972

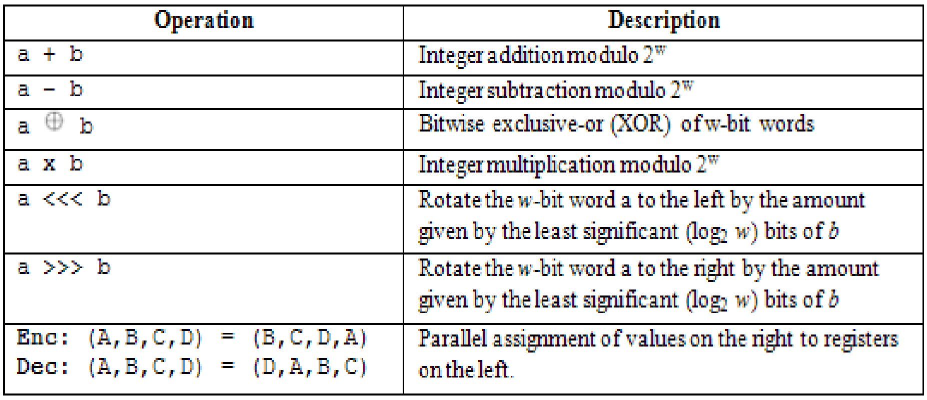
[6] Hernández, Julio César; Sierra, José María; Isasi, Pedro; Ribargorda. Arturo, "Finding efficient distinguishers for cryptographic mappings, with an application to the block cipher TEA". Proceedings of the 2003 Congress on Evolutionary Computation. 2003. http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?arnumber=1299943

[7] Hernández, Julio César; Sierra, José María; Ribagorda, Arturo; Ramos, Benjamín; Mex-Perera, J. C. "Distinguishing TEA from a random permutation: Reduced round versions of TEA do not have the SAC or do not generate random numbers". Proceedings of the IMA Int. Conf. On Cryptography and Coding 2001: 374–377

[8] Morgan Monger, “RC6: The Simple Cipher”, CS-627-0001: Cryptography Fall, 2004. http://www.rsasecurity.com/rsalabs/rc6/

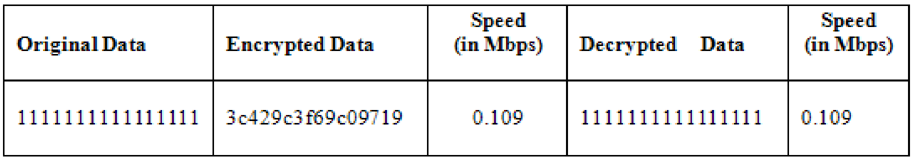
[9] William Stalling, “Cryptography and Network Security Principle and Practices”, Fourth Edition, Prentice Hall, 2005.

Table (1) RC6 Operations



Operation	Description
a + b	Integer addition modulo 2^w
a - b	Integer subtraction modulo 2^w
a ⊕ b	Bitwise exclusive-or (XOR) of w-bit words
a x b	Integer multiplication modulo 2^w
a <<< b	Rotate the *w*-bit word a to the left by the amount given by the least significant ($\log_2 w$) bits of *b*
a >>> b	Rotate the *w*-bit word a to the right by the amount given by the least significant ($\log_2 w$) bits of *b*
Enc: (A,B,C,D) = (B,C,D,A) Dec: (A,B,C,D) = (D,A,B,C)	Parallel assignment of values on the right to registers on the left.

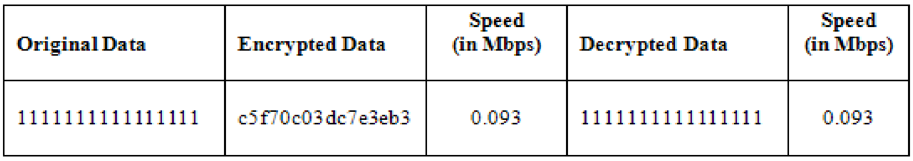
Table (2) Speed in Mbps of original TEA



Original Data	Encrypted Data	Speed (in Mbps)	Decrypted Data	Speed (in Mbps)
111111111111111	3c429c3f69c09719	0.109	111111111111111	0.109

Total Time Taken = 0.109 + 0.109 = 0.218 Mbps

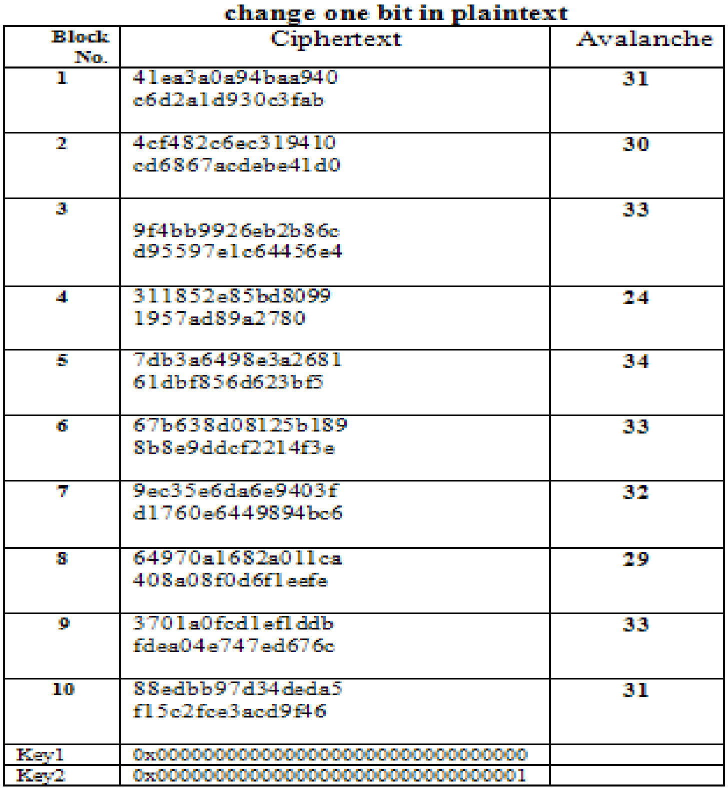
Table (3) Speed in Mbps of Modified TEA (MTEA)



Original Data	Encrypted Data	Speed (in Mbps)	Decrypted Data	Speed (in Mbps)
111111111111111	c5f70c03dc7e3eb3	0.093	111111111111111	0.093

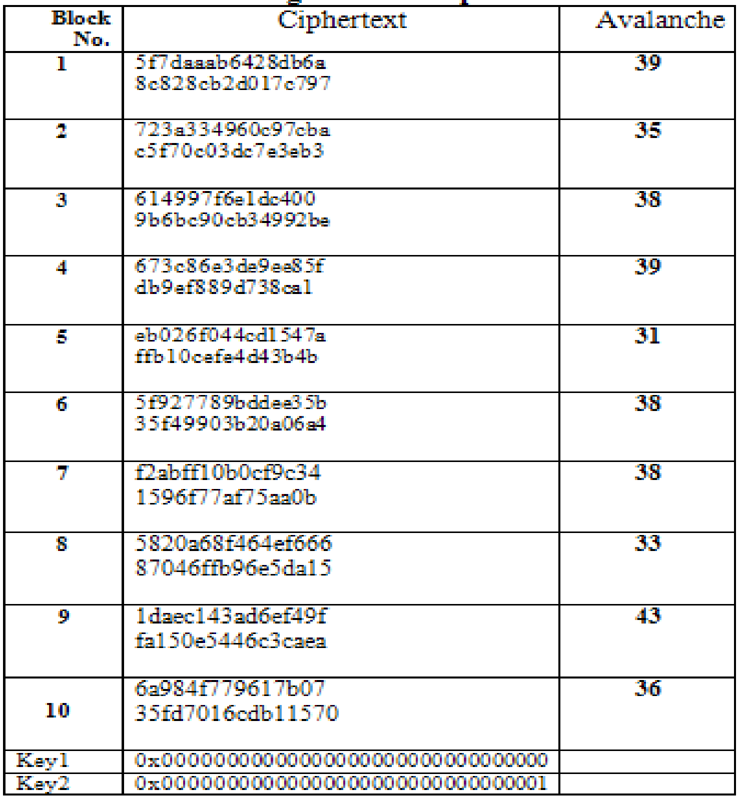
Total Time Taken = 0.093+ 0.093 = 0.186 Mbps

Table (4) avalanche effect of TEA algorithm after change one bit in plaintext



Block No.	Ciphertext	Avalanche
1	41ea3a0a94baa940 c6d2a1d930c3fab	31
2	4cf482c6ec319410 cd6867acdebe41d0	30
3	9f4bb9926eb2b86c d95597e1c64456e4	33
4	311852e85bd8099 1957ad89a2780	24
5	7db3a6498e3a2681 61dbf856d623bf5	34
6	67b638d08125b189 8b8e9ddcf2214f3e	33
7	9ec35e6da6e9403f d1760e6449894bc6	32
8	64970a1682a011ca 408a08f0d6f1eefe	29
9	3701a0fcd1ef1ddb fdea04e747ed676c	33
10	88edbb97d34deda5 f15c2fce3acd9f46	31
Key1	0x00000000000000000000000000000000	
Key2	0x00000000000000000000000000000001	

Table (5) avalanche effect of MTEA algorithm after change one bit in plaintext



Block No.	Ciphertext	Avalanche
1	5f7daaab6428db6a 8c828cb2d017c797	39
2	723a334960c97cba c5f70c03dc7e3eb3	35
3	614997f6e1dc400 9b6bc90cb34992be	38
4	673c86e3de9ee85f db9ef889d738ca1	39
5	eb026f044cd1547a ffb10cefe4d43b4b	31
6	5f927789bddee35b 35f49903b20a06a4	38
7	f2abff10b0cf9c34 1596f77af75aa0b	38
8	5820a68f464ef666 87046ffb96e5da15	33
9	1daec143ad6ef49f fa150e5446c3caea	43
10	6a984f779617b07 35fd7016cdb11570	36
Key1	0x00000000000000000000000000000000	
Key2	0x00000000000000000000000000000001	

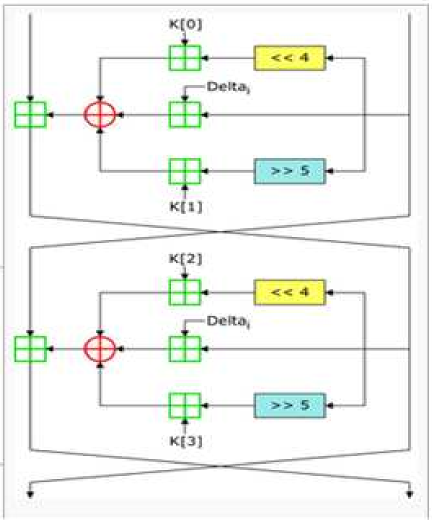


Figure 1: The TEA Algorithm

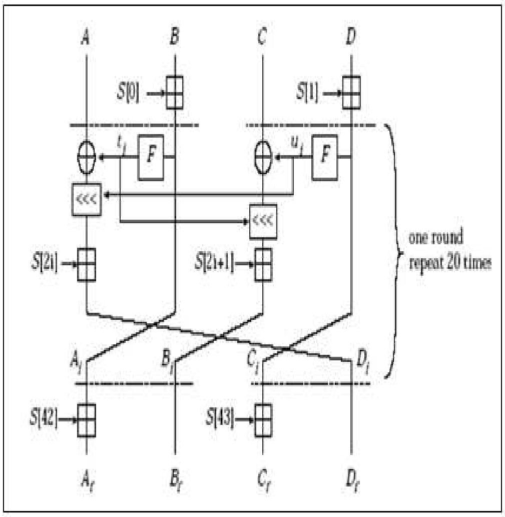
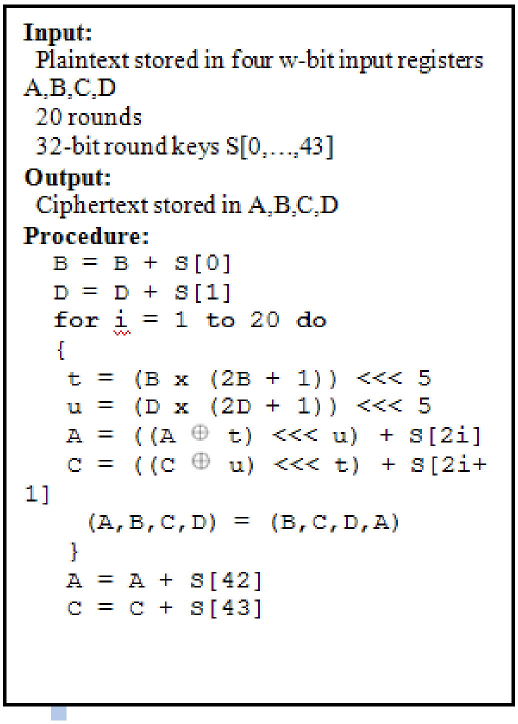


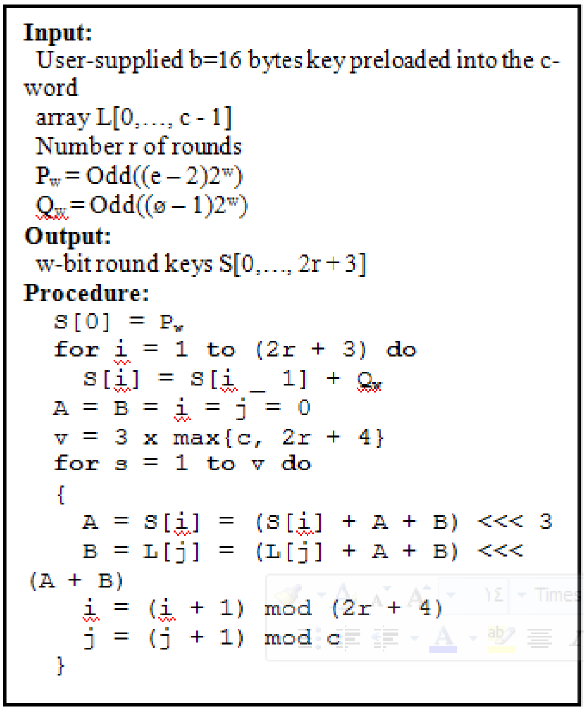
Figure 2: RC6 AES Encryption



Input:
Plaintext stored in four w-bit input registers A,B,C,D
20 rounds
32-bit round keys S[0,…,43]
Output:
Ciphertext stored in A,B,C,D
Procedure:

```
  B = B + S[0]
  D = D + S[1]
  for i = 1 to 20 do
  {
   t = (B x (2B + 1)) <<< 5
   u = (D x (2D + 1)) <<< 5
   A = ((A ⊕ t) <<< u) + S[2i]
   C = ((C ⊕ u) <<< t) + S[2i+
1]
     (A,B,C,D) = (B,C,D,A)
   }
   A = A + S[42]
   C = C + S[43]
```

Figure 3: RC6 Encryption algorithm for AES with RC6-32/20/[16,24,32]



Input:
User-supplied b=16 bytes key preloaded into the c-word
array L[0,…, c - 1]
Number r of rounds
$P_w = \text{Odd}((e – 2)2^w)$
$Q_w = \text{Odd}((ø – 1)2^w)$
Output:
w-bit round keys S[0,…, 2r + 3]
Procedure:

```
  S[0] = Pw
  for i = 1 to (2r + 3) do
    S[i] = S[i _ 1] + Qw
  A = B = i = j = 0
  v = 3 x max{c, 2r + 4}
  for s = 1 to v do
  {
    A = S[i] = (S[i] + A + B) <<< 3
    B = L[j] = (L[j] + A + B) <<<
(A + B)
    i = (i + 1) mod (2r + 4)
    j = (j + 1) mod c
  }
```

Figure 4: Key schedule with RC6-w/r/b

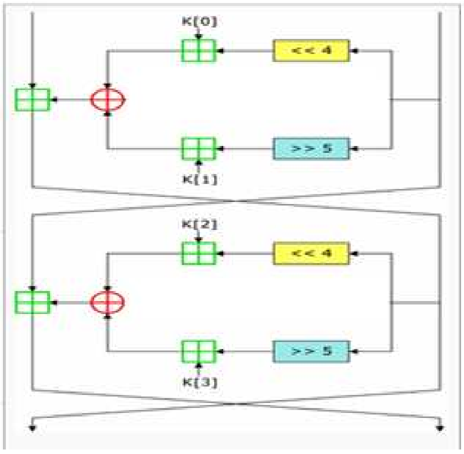


Figure 5: MTEA single round

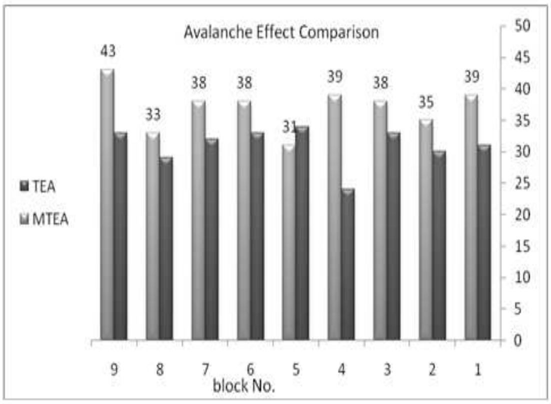


Figure 6: Avalanche Effect Comparison Times New