The history of the development of the TEA algorithm

Asst.prof.Dr.Ekhlas Khalaf Gbashi & Doaa Salam

University of technology/computer science department, Baghdad, Iraq

Abstract

As PC frameworks turn out to be more unavoidable and complex, security is progressively critical. The protected transmission evades to send and receive data, whether secret or restrictive information thought a safe channel. Most of the strategies of transferring data need a class of encryption. One of the best Encryption Algorithms is Tiny Encryption Algorithm (TEA) which is characterized by being very fast and its lightweight. Moreover, it requires only small source code. The simplicity of the key scheduling makes TEA suffering from the attacks of related key and equivalent key. In this paper we mention the modified TEA by David Wagner to eliminate the weakness, and then XXTEA is designed to improve the performance and correct the faults in the original Block TEA.

1. Introduction:

One of the important concerns in the world of portable devices, like mobiles phones or personal digital assistants (PDAs), is the security of information. Basically, cryptographic algorithms and protocols constitute the primary part of the systems which keep network transmission and data safe. In these devices, there is a challenge to achieve high performance with low consumption of power. One way to do that, is the proper selection of an algorithm for embedding as one part of the hardware [1].

The Tiny Encryption Algorithm considered among the most efficient and quickest cryptographic algorithms. This algorithm is built in the Laboratories of Cambridge University via David Wheeler and Roger Needham. It basically a Feistel cipher that applies several processes from various algebraic sets which are (ADD, SHIFT, and XOR). Using this way, it's possible to afford the identical two properties of Shannon for diffusion and confusion, these are very important for marinating the security of a block cipher, so there is no need for both of P-boxes and S-boxes. The encryption is done by a 128-bit key, where 64 data bits is encrypted every time. This algorithm has high resistant to differential cryptanalysis, beside it can reach full diffusion after 6 rounds. Therefore, when there is a difference with only one bit

in the plain text, it produces about 32-bit difference in the ciphertext. In case of using workstation computer of a desktop computer, the performance becomes very remarkable [2]. This is because use a simple logic in the process of key scheduling. This algorithm (TEA) has been attacked from the related key as well as the equivalent key. Hence, a modified key schedule is suggested to solve this issue. The Boolean function based SBox is used for the new key schedule, in order to have different round keys to TEA. In compare with the origin TEA, the new (modified) TEA has improved security [3].

XXTEA (which is TEA with corrected block cipher) is proposed to reduce the drawback in the initial block TEA [4].

Tiny Encryption Algorithm (TEA) which is designed by Roger Needham and David Wheeler at Cambridge University, considered as one of the lightweight algorithms that used in encryption. It issued for the first time in 1994 at the workshop of fast software encryption which held in Leuven. TEA "is probably the most efficient—and hence fastest—software encryption algorithm ever devised". The length of block in TEA is 64-bit and it has a key with length of a 128-bit. The length of the key is satisfying the recommendation of the latest encryption conditions. The 64-bit block is consisting from two words, each one has 32-bit. If we assume them as V [0] and V [1], then in the same way the 128-bit key will be consisting from 4 words, which are: (K [0], K [1], K [2], K [3]). The figure 1 shows the block diagram of TEA. TEA has Feistel structure, the number of rounds is 64. Each iteration is composed from double Feistel rounds. Therefore, the final iteration count will be 32. Six iterations are enough for reaching the full diffusion, but for the purpose of ensuring security, 32 iterations are used. In case of encryption time is little, it's possible to decrease the number of iterations. TEA has tiny footprint with respect to resource conditions. The souse code of TEA is very small and can be executed using most programming languages. TEA employs number symbolize as Delta namely derivative from the golden ratio where Delta = 9E3779B9 in hexadecimal [3].

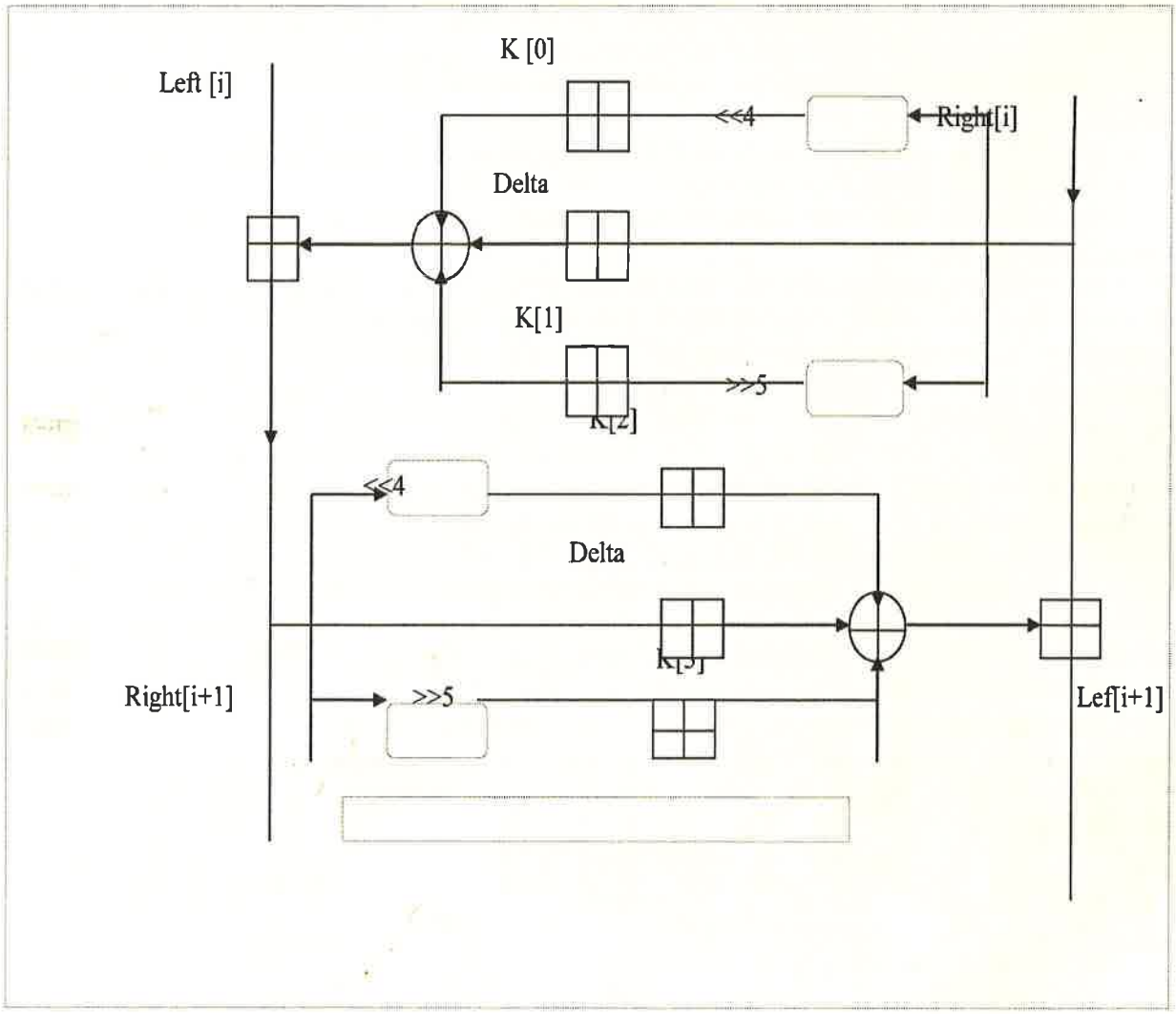

Fig1. Two Feistel rounds =1 ''cycle'' of TEA. [2]

R is considered as an input to various operation in the initial Feistel round.

1. R moves throughout a left shift of "4" next is summed to K[0]
2. Further, Adding R with Delta.
3. R moves throughout a right shift of "5" next is summed to K[1]

Moreover, the output of the procedure above is passed through XOR. Then, the result is summed with L. This output is assumed as R to the following round due to the swap [5].

1.1 Vulnerabilities and Attacks on TEA

In spite of the speed and the efficiency of TEA, but its key schedule is so simple. This key (which is 128-bit) is consisting from 4 words, each word with length 32-bit. Each one of them is symbolize as K [0], K [1], K [2], and K [3] is used as shown underneath.

If (Fiestel_Round is odd)
{Apply K [0], K [1]} else
{Apply K [2], K [3]}

This method is characterized using same K [0], K [1], K [2], and K [3], in every cycle of the 32 cycles. Therefore, in each cycle, an identical key material is applied. As a result of this weak point, TEA is exposed to related and equivalent key attacks [3].

1.2Example of TEA for one cycle

Plaintext: V [0] = 14045105, V [1] = 38207800
Cipher Key: K [0] = 27320268, K [1] = 38055350,K [2] = 28658759, K [3] = 30003515
Delta = 9E3779B9
1- convert v[0] ,v[1],k[0],k[1],k[2],k[3] into binary:
V [0] =00010100000001000101000100000101
V [1] =00111000001000000111100000000000
K [0] =00100111001100100000001001101000
K [1] =00111000000001010101001101010000
K [2] =00101000011001011000011101011001
K [3] =00110000000000000011010100010101
Delta = 10011110001101110111100110111001

2. V [1] moves throughout a left shift of 4, next is summed to value of K [0]
V [1] =00111000001000000111100000000000 →after left shift of 4 become
V [1] =10000010000001111000000000000000
+
K [0] = 00100111001100100000001001101000

01010100100111001110000100110100 0

3. V [1] is added to Delta:
V [1] =00111000001000000111100000000000
+
Delta = 10011110001101110111100110111001

01011101001000111101101011011001

4. V [1] moves throughout a right shift of 5, next is summed to value of K [1]

V [1] =00111000001000000111100000000000 →after right shift of 5 become

V [1] =00000001110000010000001111000000

+

K [1] =00111000000001010101001101010000

01110011100011001010111000100000

5- The result of the operation above is subjected to an XOR operation:

01010100100111001110000100110100 0

⊕ **01011101001000111101101011011100 1**

01110011100011001010111000100000

00010101010111000001000001100000 1

6. The output of the XOR is summed to left half of V[0]. This output will be considered as R for the coming feistel round:

00010101010111000001000001100000 1

+

V [0] =00010100000001000101000100000101

01010010110000001100001011001 01 → become new right half

V [1] new= 01010010110000001100001011001 01

V [0] new =V [1] old=00111000001000000111100000000000

Repeat the same steps in the next round with k[2] and k[3] which is considered a complete cycle .

2. Development stage of TEA:

2.1 Enhanced Tiny Encryption Algorithm (XTEA):

The Extended Tiny Encryption Algorithm (XTEA) basically is a block cipher which employs a 128-bit cryptographic key for the purpose of encrypting or decrypting data in a 64-bit blocks. Every input block is consisting from 2 components: Ln and Rn. Next, they subjected to a routine such like a Feistel network for number of rounds equal to: N, where N is normally 32. The output of a mixing function, in many of Feistel networks, is employed to one part of data by means of XOR in the form of a reversible function.

Moreover, XTEA applies integer increment modulo 2^32 while the encryption and decrement modulo 232 while decryption.

Processes used in XTEA is only XOR, increments and shifts in order for encryption. The changing in the key schedule of TEA causes the security of XTEA to be better than TEA. The key schedule of TEA is not complicated, and this may lead to expose to attacks as stated

in last section. Hence, in order to improve security of TEA, key schedule on the basis of SBox is suggested. [3]

1.1 New Key Scheduling Algorithm:

A diagram of the process of the suggested key scheduling can be shown in Fig. 2. The two primary elements, replacement comes after the left rotation of every 32-bit key. This replacement is accomplished by 4x4 SBox, that is planned by cryptographically robust Boolean Functions. The input of SBox is 4 bits, and its output is also 4 bits.

The possibility of the suggested key schedule is receiving, from the user, a cipher key of 128-bit to get round keys which are 32 keys, every key is 128-bit. These 32 keys are applied in 32 rounds of TEA [3].

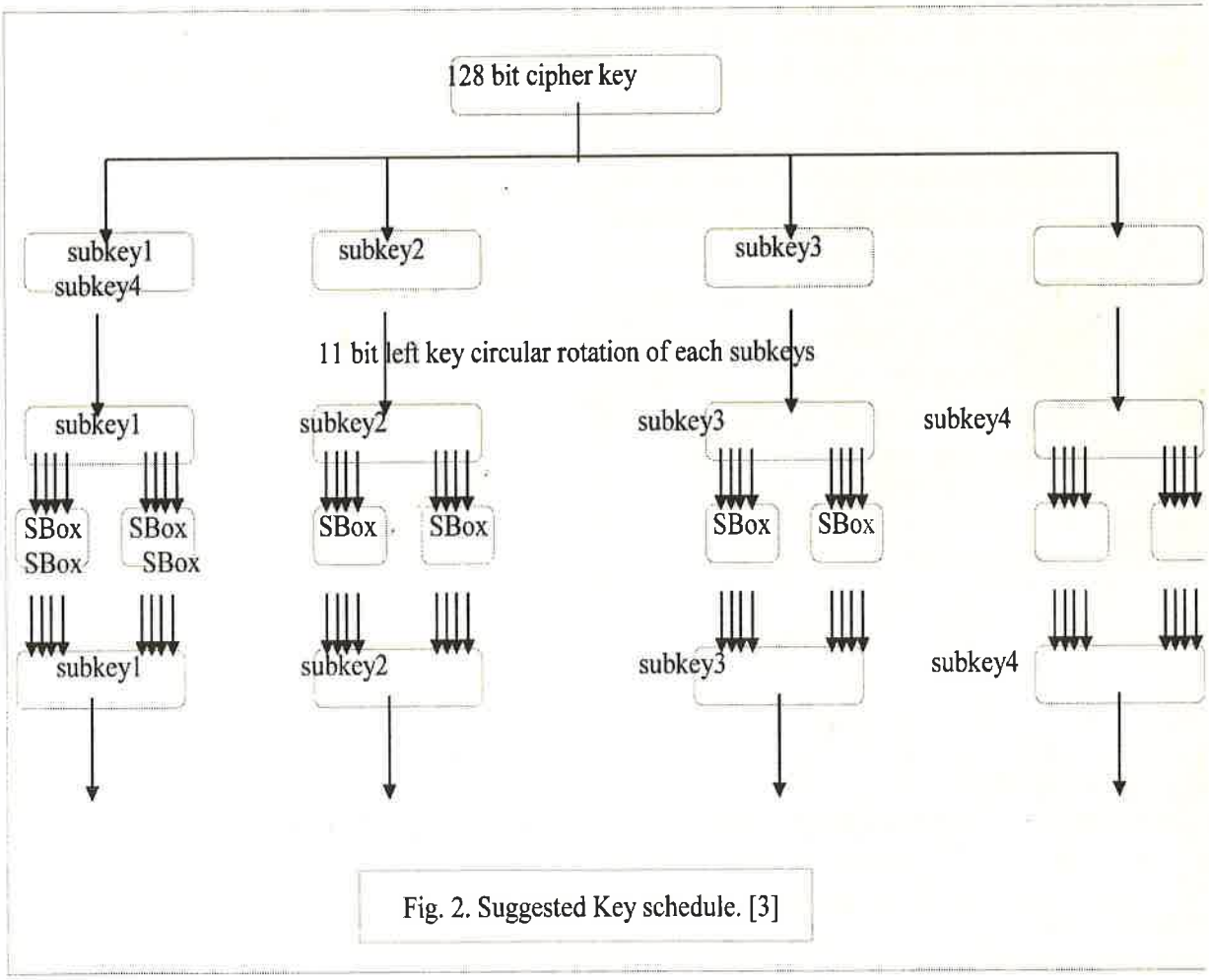

Fig. 2. Suggested Key schedule. [3]

The cipher key of length 128-bit, which is supplied by the user is separated firstly to 4 equal words, each one is 32-bit. These words rotated independently using 11-bit positions to the left direction. When the 4 words finish rotating, the 4x4 SBox is employed two times for each rotated word.

The location of each bit that transfer to the SBox is constant for each 32-bit subkey. The bit locations 0, 1, 2, 3 are applied by the first SBox, while the bit positions 28, 29, 30, 31 is used by the 2nd SBKZ. The location of SBox does not affect, this is because if they located anywhere they will change 4 bits by new other 4 bits.

At last, each TEA round uses a key with 128-bit, and this operation is continuing in the similar way, which mean that the output of an individual round considers as an input to the next round. The fig.3 shows the integration between the basic TEA and the key scheduling scheme. There is a 128-bit key (K [0] to K [3]) produced from every round of key schedule, this key is used in every round of TEA [3].

Example of on cycle of XTEA

The Plaintext: V [0] = 14045105,V [1] = 38207800
Cipher Key: K [0] = 27320268, K [1] = 38055350, K [2] = 28658759, K [3] = 30003515
Delta = 9E3779B9
1- convert v[0] ,v[1],k[0],k[1],k[2],k[3] into binary
V [0] =00010100000001000101000100000101
V [1] =00111000001000000111100000000000
K [0] =00100111001100100000001001101000
K [1] =00111000000001010101001101010000
K [2] =00101000011001011000011101011001
K [3] =00110000000000000011010100010101
Delta = 10011110001101110111100110111001

Key Scheduling Algorithm

1- At first, the user supply a cipher key with length of 128-bit. This key then separated into 4 similar words, each one with 32-bit length. Each word is independently turned by 11 bit locations to the left direction
K [0]rot ate =10010000000100110100000100111001
K [1]rotate =00101010100110101000000111000000
K [2]rotate =00101100001110101100100101000011
K [3] rotate=00000001101010001010100110000000
2- The 4x4 SBox is used two times for each word after rotation process. The locations 0,1,2,3 is used by the 1st Sbox, while the bit locations 28, 29, 30, 31 is used by the 2nd SBox for every 32 bit subkeys. The candidate Boolean Functions using for SBox, are:
f1 = x1x2 ⊕ x2x3 ⊕ x3x4 ⊕ x1x3 ⊕ x1x4
f2 = x1x2 ⊕x2x3 ⊕ x3x4 ⊕ x2x4 ⊕ x1x4
f3 = x1x4 ⊕ x2x3 ⊕ x3x4 ⊕ x2x4 ⊕ x1x3
f4 = x1x2 ⊕x3x4 ⊕ x1x3 ⊕ x1x4 ⊕ x2x4

K [0]rot ate =**1001**00000001001101000001001**1001**

0	1	2	3
X1	x2	x3	x4
1	**0**	**0**	**1**

f1 = x1x2 ⊕ x2x3 ⊕ x3x4 ⊕ x1x3 ⊕ x1x4
f1 =**1*0⊕0*0⊕0*1⊕1*0⊕1*1**
f1=1
f2 = x1x2 ⊕x2x3 ⊕ x3x4 ⊕ x2x4 ⊕ x1x4
f2=**1*0⊕0*0⊕0*1⊕0*1⊕1*1**
f2=1
f3 = x1x4 ⊕ x2x3 ⊕ x3x4 ⊕ x2x4 ⊕ x1x3
f3=**1*1⊕0*0⊕0*1⊕0*1⊕1*0**
f3=1
f4 = x1x2 ⊕x3x4 ⊕ x1x3 ⊕ x1x4 ⊕ x2x4
f4=**1*0⊕0*1⊕1*0⊕1*1⊕0*1**
f4=1

Repeat the same with k[1],k[2] and k[3] then it produces four keys ,finally 128 bit key used with one round of TEA .

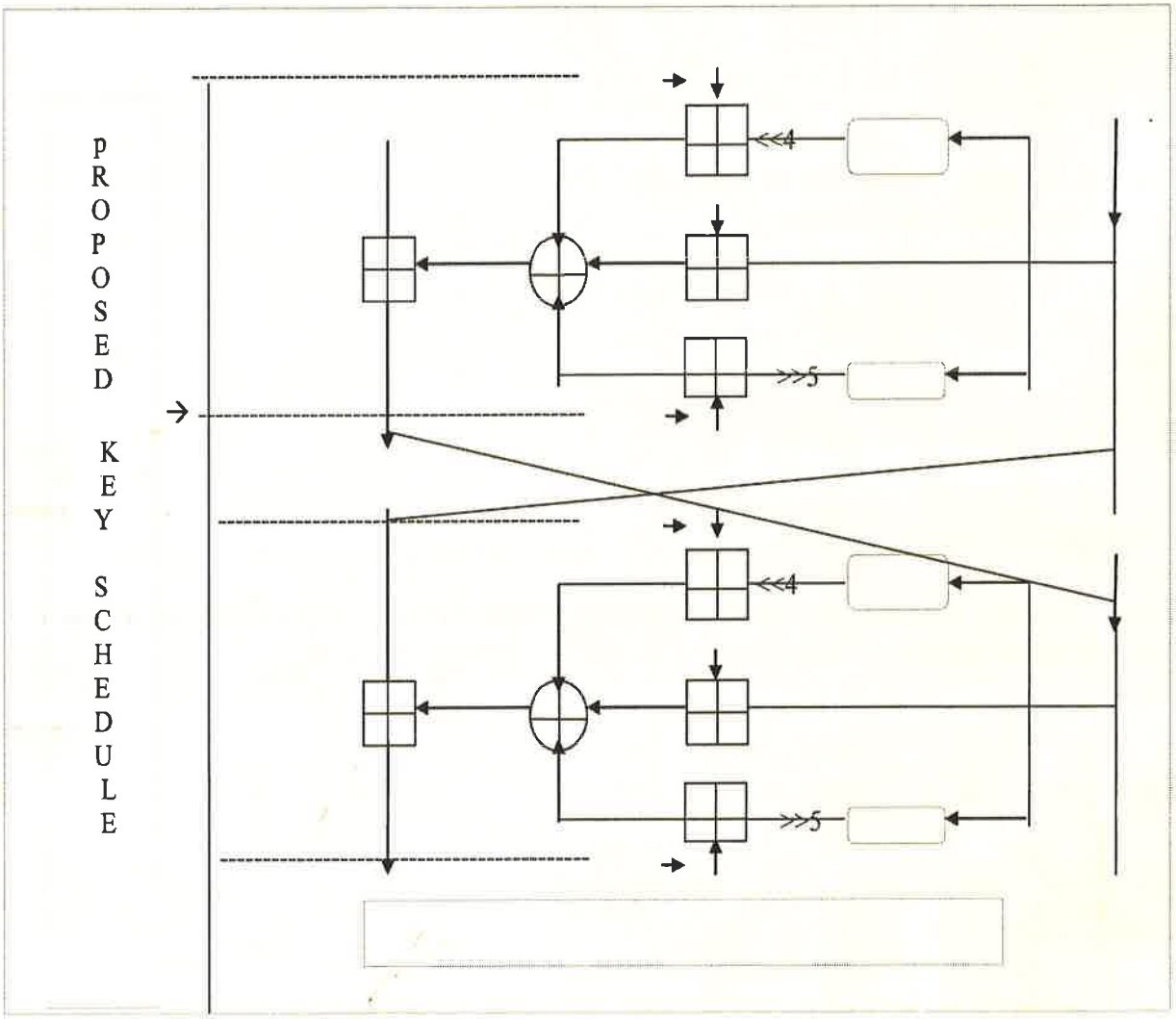

Fig. 3. Proposed XTEA with new Key schedule [3]

In case of XTEA, there is a modification in the key scheduling in order to reproduce various patterns to get mixture of data and key on an ongoing manner every round to increase the substantial confusion. The subkeys used are 4 only, with length of 32-bit, and basic increment and decrement operation come after the module 232. The logic shifts can be represented as logical left shift using four, and a logical right shift using five, besides an easy 32-bit operation using XOR.

The function f(x) is called the permutation function and can be stated by:
$f(x) = (x<<4 \oplus x >> 5) + x$.
While the generation function of the subkey can be stated by:
sum + k (sum ∨ 3), sum + k (sum >> 11 ∨ 3)
Adding acts like a chooser from the 4 subkeys k0, k1, k2, and k3 reliant on on bits 0 and 1 of the sum or bits 11 and 12. The outcomes of the variation function and generated subkey are entered operations of XOR and ADD to v0 and v1. It is noticed that the result of summation is initialized to 0 before the beginning of the calculation, and the value of delta will be constant at 0x9E3779B9. [7]

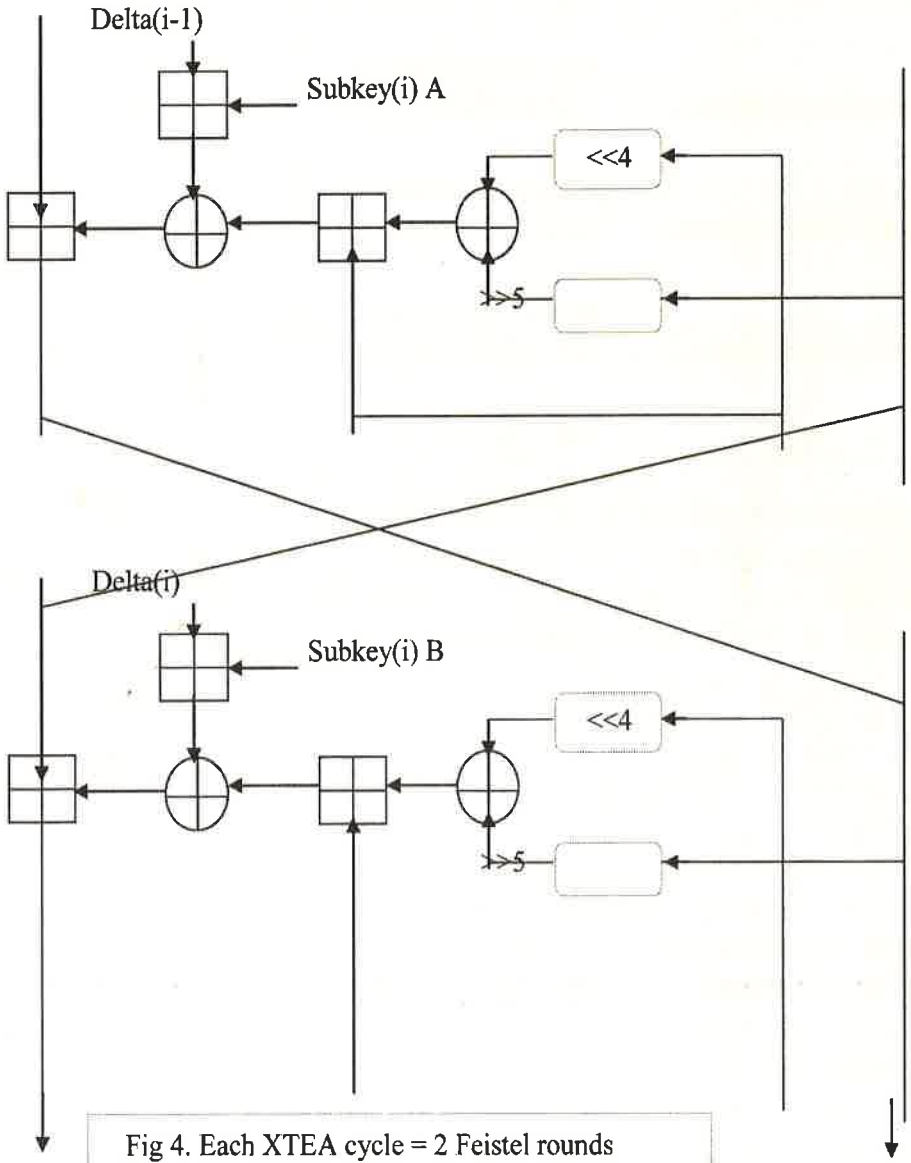

Fig 4. Each XTEA cycle = 2 Feistel rounds

Equivalent key attack: The two keys are equivalent if the plain text is encrypted with key K and this key is similar to *K cipher text. [6]

Related-key attack: If the attacker is able to watch the operation of a cipher in case of using various keys where their values are primary unidentified. Although, if the mathematical relations which linking the keys is available to the attacker in any arrangement of cryptanalysis [6].

Comparison of TEA and XTEA with the Equivalent Keys

Plain Text	Key	Cipher with TEA	Cipher with XTEA
0000000000000000	0000000800000000 0000000000000000	9327c497 31b08bbe	4f190ccf c8deabfc
0000000000000000	8000000000000000 0000000000000000	9327c497 31b08bbe	57e8c05 50151937
0000000000000000	8000000000000000 8000000000000000	9327c497 31b08bbe	31c4e2c6 347b2de
0000000000000000	0000000800000000 8000000800000000	9327c497 31b08bbe	ed69b785 66781ef3

2.2 Corrected Block TEA (XXTEA)

Corrected Block TEA (or XXTEA) can be defined as a block cipher its function is repairs the defects in the primary TEA block. This algorithm is founded in the laboratory of computer in Cambridge university by both Roger Needham and David Wheeler.

They present this algorithm in October 1998 in a technical report (not published yet). XXTEA is a block cipher featured as fixed uncompleted heterogeneous which its origin is heavy UFN (referred to unbalanced Feistel network). XXTEA algorithm runs on blocks with changing length, where some random multiple of 32 bit in length (minimum is 64-bit). The block length (or size) is determines how many full cycles are there, nevertheless six uprising to 32 for small block size are available at least.

The round function in XTEA is used in every word in XTEA primary block, then it merges with the neighboring block which lies in the far left. The rate of diffusion is small for the decryption progression, which instantly used to crack the cipher.

The modified TEA block applies more involved round functions that take advantages of instant neighbors in treatment of every word in the block. The efficiency of XXTEA is better than XTEA related to the long messages.

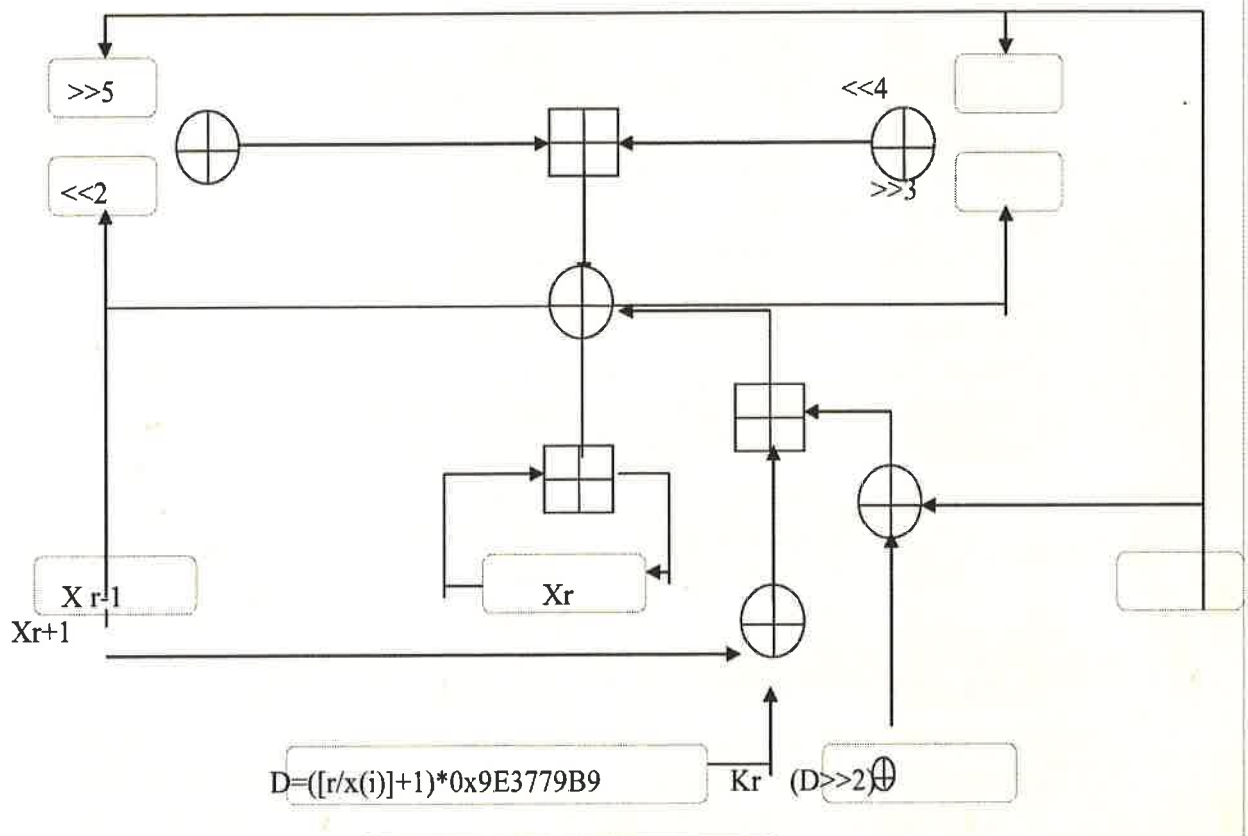

Fig 5. Cycle of XXTEA.

Comparison of TEA , XTEA ,DES and AES

	TEA	XTEA	DES	AES
Overview	Presented by Roger Needham and David Wheeler at 1994 in the laboratories of Cambridge University. It published at the workshop of Fast Software Encryption.	Presented by Roger Needham and David Wheeler in 1997 at the laboratories of Cambridge University, this algorithm was included for the first time in unpublished technological report in 1997.	Referred to Data Encryption Standard (1974). IBM organize this standard on the basis of Lucifer cipher which is the initial standard of encryption presented by NIST (National Institute of Standards and Technology).	It is an encryption standard with advanced features which is designed by Vincent Rijmen and by Joan Daemen was adopted to be the new standard of encryption in USA by October 2000. It presented by the National Institute of Standards and Technology.
	TEA (Tiny Encryption Algorithm) is featured with non-complicated structure and the simplicity in execution. Usually it need only few lines for coding.	XTEA, stands for (Enhanced Tiny Encryption Algorithm) is featured with non-complicated structure and the simplicity in execution. Usually it need only few lines for coding. It designed to improve the performance of TEA	Previously, DES was supposed as one of the robust algorithms, but this algorithm is limited in use now due to the big data and short length of keys.	Rijndael applying adjustable key length is very fast and compressed cipher. It featured with excessive elasticity due to its symmetric and parallel structure, besides its active resistance to cryptanalytic attacks.
	TEA algorithm is considered as Feistel Structured with symmetric	XTEA is using key of 128-bit length, and it considered as 64-bit block Feistel	DES is an algorithm with a symmetric key. It depends on the Feistel structure.	AES is an algorithm with a symmetric key also. It works according Feistel structure. This algorithm

Architect-ure	key. It's a block cipher which apply a 64-bit plain text through 64 rounds. It has 32 cycle in case using key length of 128-bit with changing rounds (it's better to use 64 Feistel rounds). There is no SBox, and it can use also for decryption.	cipher, with suggested 64 rounds. It differs from TEA with apparent, having more complicated key schedule and re-organize of the shifts, XOR and summations.	DES using 64-bit plain text with its block cipher, through 16 rounds with 56-bit length for the key. Initially the key having length of 64-bit (like the size of the block), but there is a single bit in each byte using as a 'parity' bit, which can not be used for the purpose of encryption. The 56-bit is changed to 16 sub-keys with 48-bit length for each of them. Moreover, it has 8 SBoxes. For decryption, the same algorithm can be used.	is a block cipher and applying 128-bit key plain text with changing 10, 12 or 14 rounds. (The Rijndael's Default number of rounds is based on the key size, and it can be calculated as key length/32 + 6) and the changing length of the key which is 128, 192, 256 bit is converted to 10 sub-keys, each one with length of 128, 192, 256 bit. It has one SBox only and can be used also of decryption.
Security	TEA and IDEA have the equals level of security. TEA has 128-bit key size, and featured with the simplicity of structure and implementation.	XTEA have an equal level of security in compare with IDEA, it also has 128-bit key size, and featured with the simplicity of structure and implementation	The 56 bit key size is determines how much DES security is strong. This key generates 7.2 x 1016 possible keys, for that reason it is so hard to create a specific key in usual threat situations. Basically, DES is very secured and it's	Security of Rijndael rest on on its adjustable nature key size permitting to a key size of 256-bit, to offer resistance contrary to several attacks on the future (collision attacks with potential quantum are calculating algorithms).

			difficult to crack.	
Advantag-e	Simplicity of structure and implementation, Normally needs only a few lines for coding and the data transmission are safe.	Supply both encryption, embedding and the transmission of data is safe.	The speed of hardware implementations of DES is very high.	Brute Force is safety (128 Bit = 2128 attempts) Very hard to breake.
Disadvan-tage	equivalent key attack and related key attacks	ETEA having equivalent keys and the related-key cann brake with 223 selected plaintexts and a time complication of 232.	DES is not secure due to a brute force attack which is possible, and DES was not planned for software so it has slow execution.	AES-128, the key can be improved with a computational complication of 2126.1 by bicliques.

Conclusions:

This paper presents a simple survey of TEA algorithm and its extensions, also a comparison between this algorithm and others is presented, this paper summarizes the rules of various cryptanalysis, the strongest attacks which face Feistel ciphers, and presents several basic collection theories. In TEA algorithm the same keys are used in all cycles, where the cycles k[0] and k[1] employed in in odd round while k[2] and k[3] are applied the even round that reduced same cipher text , but in XTEA these keys entered Sbox that changed the bits of keys where that make it more random and more sophisticated that produce different cipher text that mean with XTEA eliminate the equivalent keys and XXTEA that used too , to eliminate the weakness of original TEA .

References:

[1](Niladree De,eld) ,A Modified XTEA, International Journal of Soft Computing and Engineering (IJSCE), 2012
[2] Simon J. Shepherd, The Tiny Encryption Algorithm, Mortimer House, 37-41 Mortimer Street, London W1T 3JH, UK, 2013
[3] Chandradeo Kumar Rajak, Implementation of Modified TEA to Enhance Security, Computer Engineering Department, Defence Institute of Advanced Technology (DU), Girinagar, Pune, India,2018
[4] Elias Yarrkov, Cryptanalysis of XXTEA, May 4, 2010
[5] (Benjamin Andrews,eld), Tiny Encryption Algorithm (TEA),
[6] VIKRAM REDDY ANDEM, A CRYPTANALYSIS OF THE TINY ENCRYPTION ALGORITHM, The University of Alabama TUSCALOOSA, ALABAMA, 2003
[7] Mohamed H. Al Meer, Programmable SoC for an XTEA Encryption Algorithm Using a Co-Design Environment Replication Performance Approach, Computer Science & Engineering Department, College of Engineering, Qatar University, Doha, Qatar, 2017

3. الاعلان العالمي لحقوق الانسان لسنة 1948.
4. العهد الدولي للحقوق المدنية والسياسية لسنة 1966.
5. الاتفاقية الأوربية حول الاعتراف بالشخصية القانونية للمنظمات غير الحكومية لسنة 1986.
6. اتفاقية جنيف بشأن حماية الأشخاص المدنيين في وقت الحرب لسنة 1948.
7. مشروع لائحة تتعلق بالإغاثة الجماعية للمعتقلين المدنيين من اتفاقية جنيف بشأن حماية الأشخاص المدنيين في وقت الحرب لسنة 1949.
8. اتفاقية جنيف بشأن تحسين حال جرحى ومرضى وغرقى القوات المسلحة في البحار لسنة 1949.
9. اتفاقية جنيف لتحسين حال الجرحى والمرضى بالقوات المسلحة في الميدان لسنة 1949.
10. اتفاقية حنيف بشأن معاملة أسرى الحرب لسنة 1949.
11. النظام الاساسي للجنة الدولية للصليب الاحمر لسنة 1998.