Security aspects and comparison of block ciphers LED and TEA

Michael Appel[1], Christof Pauer[1], and Alexander Wiesmaier[1,2,3]

[1] TU Darmstadt
[2] AGT International
[3] Hochschule Darmstadt

Translation by Steven Cooper[1]

Abstract. This paper examines different encryption algorithms, which are specifically used for mobile and embedded systems on the Internet of Things(IoT). For this the Block ciphers TEA and LED will be introduced and examined with regards to their security. Afterwards they will be compared to each other and the advantages and disadvantages presented. The main focus is set on the performance and the security of the algorithms.

Keywords: Internet of things (IoT); lightweight block ciphers; LED; TEA

1 Introduction

1.1 Introduction

Through the technical progress the internet has increasingly moving into our daily lives. The increasingly smaller and cheaper expectant electronic control and communication components were installed in particular in recent years, increasingly in things of daily life. This trend is called Internet of Things (Iot). Typical fields of application are for example home automation, Security technology in the private or business environment as well as the supporting usage in the industry [30]. Because of the very high price sensitivity in this environment the focus is set on the efficiency of the used programs and algorithms. Requires an optimized algorithm for example just the half on computing time and memory, it is accordingly possible to use cheaper hardware. Extrapolated to the produced number of units a considerable amount can be saved or the IoT technology can be built-in accordingly cheaper things. Due to the growing integration of technology in the daily life and inevitably into the highly personal sphere, the claim of confidentiality and security on the collected data and the networked devices is increasing. Against this background particularly efficient algorithms have been developed, which are partly specially adapted to the used hardware. In this paper 3 block ciphers which are suitable to use on weak devices will be presented, compared and illuminated based on previously introduced attack methods of known vulnerabilities. Further the performance from the algorithm is set forth and compared with a standard process.

The paper is structured as follows: Section 2 gives a short overview about further papers which handle with similar topics as this paper. In section 3 some attack procedures on block ciphers are explained. In Sections 4 and 5, the block encryption method TEA and LED will be presented and highlights their characteristics. These encryption methods have been chosen for the analysis of this paper because they are light weighted and performant but also because they are really used and they have been successfully attacked (TEA see 5.6). LED is very useful in implementations with minimal hardware and therefore particularly in terms of interesting for the RFID sector. Also possible vulnerabilities of the previously introduced attack methods are presented. In Section 6 the block ciphers are compared with each other under performance and efficiency points as well as with AES aspects. For better alignment this paper contains compare values of the block encryption method KATAN. Section 7 gives a short conclusion. For better alignment this paper contains compare values of the block encryption method KATAN.

1.2 Mathematical notations

This paper uses the following mathematical operators:

Integer addition: The addition of two integer numbers modulo $2^n$is written down as $x \boxplus y$. Where $x, y \in \mathbb{Z}_{2^n}$.The value of n results from the context.

Exclusive or (xor): $x \oplus y$

Bitwise Shift: The logical shift from x to y bits is written down as $x << y$ (to left). The logical shift from x to y to right is written down as $x >> y$.

2 Related Work

Due to the notoriety grades of TEA and LED there exist many papers which also deal with these algorithms from different points of view. First to call are the papers which generally deal with these encryption methods. They each illustrate one of the algorithms and concentrate on certain properties in different detail degrees. For the Tiny Encryption Algorithm (TEA) the following papers are cited as examples: "TEA, a Tiny Encryption Algorithm"[24], "Tiny Encryption Algorithm (TEA)"[2] and "The Tiny Encryption Algorithm (TEA)"[29]
The second relevant group are the papers which have a special focus on the lightness and therefore the suitability of the algorithms in very small, computationally weak and cheaper Hardware. Here are specially the papers: "Design and Implementation of Low Power Hardware Encryption for Low Cost Secure RFID Using TEA"[14] and "Hardware Implementation of a TEA-Based Lightweight Encryption for RFID Security" [15] to call.
The last group are the papers which consider the algorithms under the security aspects and describe vulnerabilities and possible attacks. These include inter alia

the following: "Related-key rectangle attack on 36 rounds of the XTEA block cipher"[18] and "Meet-in-the-Middle Attacks on Reduced-Round XTEA"[22].

3 Attacks

In this section different types of attacks on block ciphers will be shortly described. These attacks play a more or less important role for the in this paper handled ciphers and will be taken up again in the security section. The attacks are: brute-force, linear cryptanalysis, algebraic cryptanalysis, differential cryptanalysis, related-key attacks, meet-in-the-middle attacks, side-channel attacks and combinations of these methods.

Brute-force. A brute-force attack tests systematically all possible keys on the cypher text. It is assumed that the attacker dont have any prior knowledge about the keys that are more probably than other keys. This type of attack is often from minor importance because it is uneconomical to decrypt the cypher text with all possible keys. The complexity of a brute-force attack act as reference for other attacks.
Linear cryptanalysis This attack requires a known-plaintext attacker ahead, i.e. the attacker knows the relative cipher text to a certain plaintext. The idea of the attack is to find linear equations for parts or. single operations of the cypher. The equations try to determine plain text bits, cipher text bits and key bits pairs with a better probability than $\frac{1}{2}$. For these attacks it is an important performance factor how many plaintext ciphertext pairs are needed.
Algebraic cryptanalysis This attack has the same objective as the linear cryptanalysis, but instead of only linear equations also polynomial equations of any degree can be used. An often problem in this context is there is no exact complexity and because of that it is necessary to use other metrics like runtime on a specific test environment.
Differential cryptanalysis. The differential cryptanalysis is a chosen-plaintext attack, i.e. the attacker can encrypt a chosen plaintext. To accomplish the attack pairs $(\Delta X, \Delta Y)$ are compared, whereby ΔX and ΔY are in each case the difference (e.g. XOR) of two values, e.g. the difference of two inputs and outputs of the algorithm. Goal of this analysis is to classify certain keys more likely.
Related-key attack. At a related-key attack it is assumed that the attacker knows not only the cipher text of the originally keys K but also the decryption with key K' which are derived from K i.e. $K' = f(K)$.
Meet-in-the-middle attacks (MITM). MITM attacks assume at least one known pair of plain- and cipher text (known- or. Chosen-plaintext). At the first step the attacker tries to filter keys i.e. the key space is limited. At the second step the right key is searched using brute-force or another attack. More details about this attack technique can be found for example in Takanori Isobe and Kyoki Shibutanii[13].
Side-channel attacks. A side-channel attack doesnt attack the algorithm itself but the physical implementation. The attacker tries for example of the duration or the power consumption of certain operations to infer information. Also the

attacker can try to specifically tilt bits for example due to manipulate the applied voltage.

4 LED

Light Encryption Device is a symmetric block cipher which was published from Guo et al.[11] in 2011. The cipher is lightweight and can efficiently be implemented in hardware. Because of these properties the authors suggest that LED is basically suitable for encryption and decryption in the IoT area. A concrete use case is the secure storage and transmission of RFID tags.

4.1 Encryption

LED uses a block size of 64 bits. The key length is 64 bit (LED-64) or 128 bit (LED-128). Even key length between 64 bit and 128 bit are basically possible for example 80 bit. In this case the remaining bits will be padded with the prefix of the key (padding). We call the actual bytes of the blocks to be encrypted as a `State`. The state and the keys will be written down as a (4×4) matrix. The matrix entries of the state and key are respectively 4 bit blocks (Nibble) and represents elements of the body $\mathbb{F}_{2^4}$. With a key length of 128 bit, there are accordingly 2 matrices, each with 64 bit. A round consists of 4 sub-steps: `AddConstants`, `SubCells`, `ShiftRows` and `MixColumnsSerial`. The number of rounds are 32 (LED-64) or 48 (LED-128). Below each sub-step of a round will be described.

AddConstants. At the beginning of each round the operation AddConstants is performed. For this purpose, the State with the following matrix is added bitwise using XOR:

$$\begin{pmatrix} 0 & (rc_5||rc_4||rc_3) & 0 & 0 \\ 1 & (rc_2||rc_1||rc_0) & 0 & 0 \\ 2 & (rc_5||rc_4||rc_3) & 0 & 0 \\ 3 & (rc_2||rc_1||rc_0) & 0 & 0 \end{pmatrix}$$

The bit vector $(rc_5, rc_4, rc_3, rc_2, rc_1, rc_0)$ is initialized with 0 before the first round. Before each addition the matrix on the last round is taken, the bit vector is shifted by one positon to the left and rc_0 is set to $rc_5 \oplus rc_4 \oplus 1$.

SubCells. In this step every Nibble of the state will be replaced by another (Sbox). The Sbox was borrowed by the block cypher `PRESENT` [3].

Table 1. `PRESENT` Sbox.

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[x]$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

ShiftRows. For $i = 1, 2, 3, 4$ each i-th is shifted cyclical $i - 1$ positions to the left.

MixColumnsSerial. Each column vector v of the state is replaced with $M \cdot v$. To multiply the elements, the polynomial $X^4 + X + 1$ is used.

$$M = \begin{pmatrix} 4 & 2 & 1 & 1 \\ 8 & 6 & 5 & 6 \\ B & E & A & 9 \\ 2 & 2 & F & B \end{pmatrix}$$

After every step (4 rounds) and at the start of the encryption is also still the operation `addRoundKey` performed. In addition to that the key is added to the state by use of XOR. In the case of LED-128 the two keys switch after each step. The name round key is misleading at this point because the key never changes. An outline of the encryption process is shown in Fig. 1.

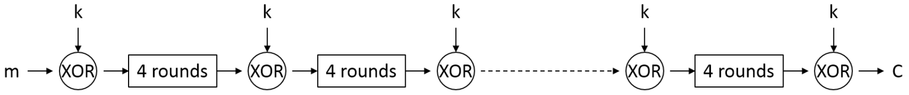

Fig. 1. LED encryption

4.2 Security

LED is very similar to AES with regard to the rounding operation. The round number is comparatively set high with 32 (LED-64) or 48 (LED-128). AES-128 uses 10 rounds. However, should be noted that AES adds the round key each round. LED uses the operation AddConstants instead. Therefore, one step (4 rounds) in LED is equivalent to 4 rounds single-key AES. This comparison is drawn because the security for 4 rounds single-key AES was further explored [21][6].

In the following, an overview of the attacks is given that exists on LED[4] are given. Vincent Grosso et al.[10] research algebraic attacks on LED. Evaluated is the time needed for the key recovery with different number of key bits. At full number of rounds, the key can be recovered in about 5 minutes[5] when 16 of 64 key bits are unknown. Result for the case that more than 16 key bits are unknown are not provided. It is also noted that the attack is only an advantage over a brute-force attack brings if at least 9 key bits are unknown.

In the papers by Xinjie Zhao et al.[32] and Philipp Jovanovic et al.[16] algebraic attacks are combined with side-channel attacks. Both attacks require that the same plain text can be encrypted with the same key twice. At the second

[4] The given values refer to `LED-64`, if not otherwise specified.

[5] A statement about the used test environment is not taken.

encryption in the 30 round after the operation SubCells it is attempted to produce an error in the first entry of the state so that at this entry is a random value. Based on the difference of the outputs and the inverse rounds a system of equations is formed with that the key space of $2^{19} \sim 2^{25}$ or $2^{6} \sim 2^{17}$ can be reduced [16][32].

Takanori Isobe und Kyoji Shibutani[13] research MITM attack on lightweight ciphers. It is about a chosen-plaintext attack with the objective to recover the key. In the case `LED-64` rounds are attacked where 2^8 plain-cipher-text pairs and 2^{56} encryptions are needed. At `LED-128` are 16 rounds at a time complexity of 2^{112} and a data complexity of 2^{16} achieved.

Mendel et al.[19] describe differential attacks in the single- and related-key context. Objective of the attack is the key-recovery. In the related-key attack the authors succeed to attack in 16 rounds at $2^{62.7}$ encryptions. However, the attack also has with $2^{62.7}$ a high data- and memory complexity. In the paper is also an attack shown on `LED-128` provided that K_0 will guess (data complexity 2^{64}) the complete key $K_0||K_1$ at 2^{96} encryptions can be recovered.

That in 2015 published paper by Ivica Nikoli et al.[21] attacks with a combination of MITM and differential crypto analysis the so far most rounds (without side-channel) with 20 (`LED-64`) or 40 (`LED-128`).In contrast to the previously noted attacks its objective is not the key-recovery. Instead at a successful attack the attacker can distinguish a randomly permutation of the 2^k possible permutations of the used key (distinguish attack). This attack is less threatening than a key recovery but on this basis with less complexity further attacks can be accomplished. The results are as follows. In the case `LED-64`20 rounds at a time complexity of $2^{60.2}$ and a data complexity of $2^{61.5}$ are attacked. For `LED-128` are 10 rounds with $2^{60.3}$ encryptions and a memory complexity of 2^{60} affected.

A summary of the attacks is shown in table 2[21]. In addition to the complexities, the attack conditions (single-key (SK), related-key (RK), chosen-key (CK)), the number of attacked rounds and the effect of the attack (key-recovery (KR), distinguisher (D)) listed.

Table 2. Attacks on LED[21]

Chiffre	Attack	Type	Rounds	Time	Data	Memory	Ref
`LED-64` (32 rounds)	MITM (SK)	KR	8	2^{56}	2^{8}	2^{11}	[13]
	Linear/Differential (CK/RK)	KR	16	$2^{62.7}$	$2^{62.7}$	$2^{62.7}$	[19]
	Linear/Differential (CK)	D	15	2^{16}	–	2^{16}	[11]
	MITM/Differential (CK)	D	16	$2^{33.5}$	–	2^{32}	[21]
	MITM/Differential (CK)	D	20	$2^{60.2}$	–	$2^{61.5}$	[21]
`LED-128` (48 round)	MITM (SK)	KR	16	2^{112}	2^{16}	2^{19}	[13]
	Linear/Differential (CK/RK)	KR	24	2^{96}	2^{64}	2^{32}	[19]
	Linear/Differential (CK/RK)	D	27	2^{16}	2^{16}	2^{32}	[11]
	MITM/Differential (CK)	D	32	$2^{33.5}$	–	2^{32}	[21]
	MITM/Differential (CK)	D	40	$2^{60.3}$	–	2^{60}	[21]

5 Tea family

5.1 Introduction TEA

The Tiny Encryption Algorithm (TEA) was developed with the objective to design a high performance and mathematical not to complicated encryption algorithm which in particular also for use on low-performing small computers in the IoT environment is. The algorithms of the TEA-family are variants of a Feistel Cipher and thus block ciphers. TEA encrypts 64 bit blocks which are directly split into 32 bit blocks. The classical TEA algorithm uses a 128-bit length key. TEA is a round based encryption method. The number of the used rounds are variable but 32 Tea cycles are recommended. Due to the symmetrical construction of the encryption algorithm (see point 5.3) is one cycle in TEA equivalent to two Feistel rounds. [2]

The algorithm exceeds the performance of DES (see point 5.7) and can be implemented in all programming languages. For many common programming languages exist reference implementations which can be used with little effort. With a strength of 32 cycles is the test implementation 60% faster than the reference implementation with 56-Bit DES. The encryption strength of TEA can be further increased by increasing the encryption cycle. [24]

5.2 "The golden number"

To counter attacks which try to exploit the symmetric of the encryption rounds it is a frequent practice by some encryption methods to include golden numbers at each round. This has the effect that there are no bits which do not change in sequential rounds. The classic golden number is defined as: [25]

$$\frac{1+\sqrt{5}}{2} \tag{1}$$

TEA uses a derived constant from the golden number: [24]

$$(\sqrt{5}-1)2^{31} \tag{2}$$

This constant initialized the variable delta and equates to a rounded integer: [24]:

$$delta = 2654435769_{10} = 9E3779B9_{16} \tag{3}$$

The mathematical definition of the constant should counter the suspicion that it is not a random number but a conscious weakening of the algorithm installing a backdoor. In cryptography it is for this reason a frequent practice not to use hardcoded random chosen numbers but to generate them by a simple comprehensible mathematical operation.[25]

5.3 Encryption algorithm

As already described under 5.1TEA is a block-cipher encryption method which can only encrypt 64 bit blocks. To encrypt the 64-bit block gets split in two 32 bit blocks. One block named L (left) the other R (right). The blocks get interchanged after each encryption round. The 128-bit key gets split in 4 sub keys and named with K[0-3]. The first 32 bit are in key K[0], the second 32 bit are in key K[1] etc. the encryption steps are shown in 2 and get formally introduced under section 5.5. For further details, you can also look into the mentioned sources. [2]

5.4 TEA encryption routine characteristics

The TEA encryption algorithm has the following characteristics:

- As usual for a Feistel cipher every round I has 2 inputs Left(i) and Right(i) from the second round it is in each case the output of the opposite side from the round before. In each case the variables will be initialised with one half of the block to be encrypted.
- The in every round used key K[i] is a part of the 128 Bit long key K.
- The constant delta will be initialised with a golden number derived constant and ensures that the partial keys generate different ciphers and have no relevant cryptographic significance. (see section 5.2).
- The encryption algorithm dont use random numbers i.e. identical text by the same key leads to the identical cipher.

5.5 Formal definition of the encryption

To clarify the functionality are hereinafter the encryption functions of the TEA block cipher listed. Figure 2 shows the encryptions steps graphically. As introduced in section 5.3 the respectively part of the used key is named K[0-3] and the respectively part of the current block to be encrypted is named with Right(i) or. Left(i). Delta[i] is a modification of the under section 5.2 introduced golden number. As described delta is defined as follows:

$$delta = (\sqrt{5} - 1)2^{31} = 2654435769_{10} = 9E3779B9_{16} \tag{4}$$

One TEA round consists of two Feistel rounds and for that reason the control variable i increases by 2 each round. The variable i indicates the Feistel rounds. In one TEA round the following operations were executed:

$$delta[i] = (i + 1)/2 * delta \tag{5}$$

$$Left[i + 1] = Right[i] \tag{6}$$

$$Right[i + 1] = Left[i] \boxplus F(Right[i], K[0, 1], delta[i]) \tag{7}$$

$$Left[i+2] = Right[i+1] \tag{8}$$

$$Right[i+2] = Left[i+1] \boxplus F(Right[i+1], K[2,3], delta[i]) \tag{9}$$

$$F(M, K[j,k], delta[i]) = ((M << 4) \boxplus K[j]) \oplus (M \boxplus delta[i]) \oplus ((M >> 5) \boxplus K[k]) \tag{10}$$

Equation 6 and 7 form the first step of a TEA round (the first Feistel round). The equations 8 and 9 close the TEA round (the second Feistel round). F() referred the so called round function (equation 10) which contains the significant steps of the cryptographically operations. The function is always the same but get called with different parameters each Feistel round.

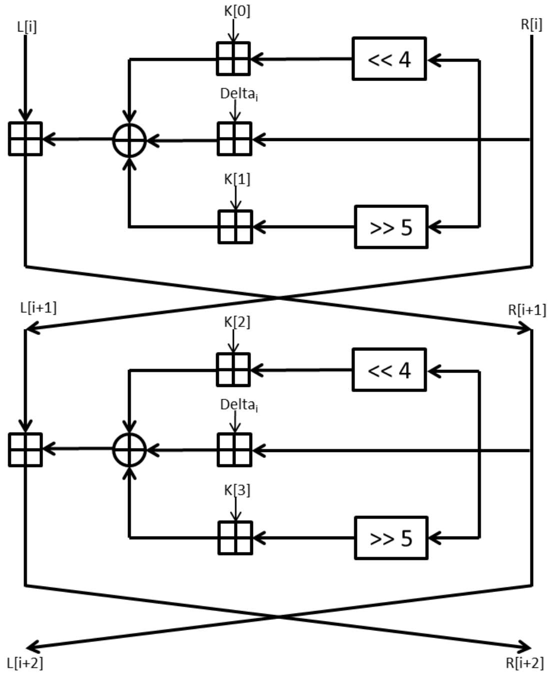

Fig. 2. Encryption steps TEA, Source: Based on [27]

5.6 Vulnerabilities of TEA

TEA has been handled in some crypto analysis and examined for vulnerabilities. The current known vulnerabilities of the original TEA implementation are:

- Hash collisions used as hash functions: TEA wasnt developed for being used as a hash function and does not meet the central condition of preimage resistance for cryptographically hash functions. That means it is possible with comparatively little effort to find to a given hash value Y an input value X its hash value after using the hash function also maps to Y. Therefore, specifically collisions can be calculated to a given hash value. [29] [26]
- Key cracking: Vulnerability for simple attempts of keys (brute force). In combination with a known plaintext-ciphertext pair the necessary iterations strongly decrease and the efficiency of the attack strongly increase.[29]
- Equivalent keys: Because of a constructional vulnerability at the TEA encryption algorithm every key to decrypt a cipher has 3 equivalent keys which can although be used to decrypt the cipher. This means that the effective key space of a 128 Bit long key is reduced to 126 Bit.[29]

The following describes detailed and exemplarily the equivalent key vulnerability of the TEA algorithm depended on [1] and derived the mathematical backgrounds. If two different keys (K and K) in an encryption system with identical plaintext generate the identical cipher both keys K and K were called as equivalent. Following equation express this:

$$E_K(T) = E_{K'}(T) \tag{11}$$

Conversely that means that a with K encrypted cipher can be decrypted with K:

$$D_{K'}(E_K(T)) = T \tag{12}$$

At a good encryption system the claim should be that there are no equivalent keys. The number of the equivalence classes of the ciphers are in this case 2^k whereby k is the key length in Bit. For TEA with a 128 Bit key length it should be $2^{128} \approx 3,4 * 10^{38}$different equivalence classes. Analysis show that in TEA are only 2^{126} different equivalence classes with a key length of 128 Bit. In this case there are only 2^{126}differentiated from each other cyphers by a given plaintext T. For every cipher there are 4 each other equivalent keys $K_0 ... K_3$ by which the plaintext can be decrypted. The following example illustrates this:

$$\forall a, b \in \mathbb{Z}_{2^{32}} \tag{13}$$

$$2^{31} \boxplus 2^{31} = 0 \tag{14}$$

$$a \boxplus 2^{31} = a \boxplus 80000000_h \tag{15}$$

This means:

$$a \boxplus b = (a \oplus 80000000_h) \boxplus (b \oplus 80000000_h) \tag{16}$$

In this way the round function of TEA can be manipulated:

$$F(M, K[j, k], delta[i]) = F(M, (K[j] \oplus 80000000_h, K[k] \oplus 80000000_h, delta[i])) \tag{17}$$

Every 128 Bit key K0K3 has three equivalent keys in the form of:

$$(K[0], K[1], K[2] \oplus 80000000_h, K[3] \oplus 80000000_h \tag{18}$$

$$(K[0] \oplus 80000000_h, K[1] \oplus 80000000_h, K[2], K[3]) \tag{19}$$

$$(K[0] \oplus 80000000_h, K[1] \oplus 80000000_h, K[2] \oplus 80000000_h, K[3] \oplus 80000000_h) \tag{20}$$

So that a 128 Bit key with the TEA encryption has only a key space of 126 Bit ant thereby the security from a 126 Bit key. [1]

5.7 Performance

As described under section 5.1 the primary goal of developing TEA was to achieve a high performance. In an exemplarily test under a Java environment TEA was 18 times faster than the Java provided DES implementation. Further tests have shown that TEA (128 Bit, 32 iterations) are 60% faster than 56 Bit DES and 4 times faster than 168 Bit 3DES. Because for the following described block versions Block TEA and XXREA it is explicitly recommended by the authors to use larger data blocks a further performance increase is expected.[29]

5.8 Further development XTEA (eXtended TEA)

The XTEA (eXtended TEA) algorithm is a further development of TEA and corrects et al. the under section 5.6 described vulnerability of the equivalent keys. Like TEA works XTEA with 64 Bit blocks and a 128 Bit key length. Recommended are also 64 encryption rounds. [18] The improvements compared to TEA were achieved due a more complex key management and a change of the Shift, XOR and addition operations. [28] Due to the current state of research even XTEA isnt an encryption method without vulnerabilities. It exists descriptions of successful attacks against XTEA due exploitation of a related key vulnerability[18]. The XTEA algorithm was aware weakened due a partly significant decrease of the encryption cycles. It can however be assumed that due an appropriate greater effort an attack is also applicable at the recommended 64 encryption rounds. These attacks are described inter alia in [17], [18] and [22].

5.9 Modification Block TEA

Block TEA was published simultaneously with XTEA and differs only slightly technically from XTEA. In contrast to XTEA Block TEA dont need a fixed block size but it can also work with blocks of any size. This means that Block TEA dont need an operation mode to ensure confidentiality and authenticity. Block TEA is applied directly to the entre message. Internally the round function (see 5.3) is iteratively and cyclically applied to the entire message The used round function is identical to XTEA. So that Block TEA has the same vulnerabilities as XTEA. [22] [28]

5.10 Further development Corrected Block TEA (also referred as XXTEA)

Corrected Block TEA or XXTEA is an in 1998 published further development of Block TEA. As Block Tea it does not have a fixed block size and can be applied to the entire message. The goal to develop XXTEA was to correct the known vulnerabilities of Block TEA. For this some changes have been made in the round function. The reference implementation of XXTEA Correction to xtea is available at [23]. Also for XXTEA it already exists documented successful attacks. The paper Cryptanalysis of XXTEA [31] describes a successful chosen plaintext attack with 2^{59} plain- ciphertext pairs.

6 Comparison of the algorithms

To compare the software and hardware implementation of the ciphers TEA and LED the crypto system KATAN[4] (or KTANTAN) is listed too. The function meadow of KATAN is strongly different to LED but for software implementations values of the same magnitude are expected. As for LED no software implementations are known, KATAN is used instead. Accordingly, we have found no reprasentative hardware implementation of TEA and use also the cipher KATAN instead. The block ciphers LED and KATAN are optimized for hardware while TEA and the reference implementation AES are optimized for usage in software implementations. By the development of KATAN and LED it was decided that they are efficiently in hardware i.e. that they can be realized with few hardware elements. Due the focus on the hardware implementation the algorithms KATAN and LED have an appropriate worse efficiency in software implementations. Table 3 shows software implementations of AES, TEA and KATAN. Striking is the significantly higher (factor 10) power consumption of KATAN to AES and TEA. The power consumption correlates with the number of the needed CPU cycles for decryption and encryption. Overall TEA is worse than AES i.e. the numbers of need CPU cycles, the power consumption and the throughput are significantly better at AES than TEA. The values in the table should be noted that the block size of AES is twice the size as TEA or LED and KATAN-64 and for that the values in the columns (CPU-)cycles and power consumption have accordingly to be normalized. Striking is the value 0 byte RAM at the TEA implementation from Thomas Eisenbarth et al.[8]. Unfortunately the source has no justification for that value. We assume that it was worked exclusively with the CPU registers and these were not valued as memory (RAM).
Table 4 compares hardware implementations of KATAN, AES and LED. AES needs a significantly larger number of gates but also have a performance advantage to the compare algorithms. LED and KATAN use a similar number of gates. The most compact implementation is possible with KTANTAN32. KATAN has a significant performance advantage to LED, this advantage increases again with increasing block sizes. An advantage of LED is that the key length is flexible and can be adjusted to the desired security level.
Table 5 shows summarized different attacks on TEA and LED. There are on

Table 3. Comparison of the software implementations (at 4MHz)

Cipher	**Block**	**Key**	**Code**	**RAM**	**Cycles**	**Cycles**	**Throughput**	**Energy**
	Size	**Size**	**Size**					
	[bits]	[bits]	[bytes]	[bytes]	[enc+key]	[dec+key]	[Kbps]	[μJ]
AES [7]	128	128	1659	33	4557	7015	-	19.2
AES [8]	128	128	2606	224	6637	7429	77.1	-
TEA [7]	64	128	648	24	7408	7539	-	30.3
TEA [8]	64	128	1140	0	6271	6299	40.8	-
KATAN[7]	64	80	338	18	72063	88525	-	289.2

Table 4. Comparison of the hardware implementations[11] [4]

Cipher	Key	Block	Cycles	Throughput	Structure	Area
	length	size	per Block	at 100 KHz (Kbps)	size	GE
			flexible keys			
AES-128 [12]	128	128		0.08	0.13	3100
AES-128 [9]	128	128	1032	12.4	0.35	3400
AES-128 [20]	128	128	226	56.6	0.13	2400
KATAN32	80	32		12.5	0.13	802
KATAN48	80	48		18.8	0.13	927
KATAN64	80	64	255	25.1	0.13	1054
LED-64	64	64	1248	5.1	0.18	966
LED-80	80	64	1872	3.4	0.18	1040
LED-96	96	64	1872	3.4	0.18	1116
LED-128	128	64	1872	3.4	0.18	1265
			fixed keys			
KTANTAN32	80	32		12.5	0.13	462
KTANTAN48	80	48		18.8	0.13	588
KTANTAN64	80	64	255	25.1	0.13	688
LED-64	64	64	1280	5.13	0.18	688
LED-80	80	64	1872	3.4	0.18	695
LED-96	96	64	1872	3.42	0.18	700
LED-128	128	64	1872	3.42	0.18	726

the one hand attacks listed which attack so far the most rounds (apart of side channel attacks) and on the other hand there is a MITM attack which can be applied on LED and XTEA[13]. At XTEA the MITM attacks the so far most rounds.

7 Conclusion

To conclude, it can be said that the background for what the algorithm is developed is very important. Focusing on the hardware implementation of LED and KATAN reflects very clearly in the performance data. For the TEA family as software algorithm there are no data regarding a hardware implementation. Due

Table 5. Attacks on (X)TEA and LED compared

Cipher	Attack	Type	Rounds	Time	Data	Memory	Ref
`LED-64`	MITM (SK)	KR	8	2^{56}	2^{8}	2^{11}	[13]
(32 rounds)	MITM/Differential (CK)	D	20	$2^{60.2}$	–	$2^{61.5}$	[21]
`LED-128`	MITM (SK)	KR	16	2^{112}	2^{16}	2^{19}	[13]
(48 rounds)	MITM/Differential (CK)	D	20	$2^{60.2}$	–	$2^{61.5}$	[21]
`XTEA` (64 rounds)	MITM (SK)	KR	29	2^{124}	2^{45}	2^{4}	[13]
`TEA` (64 rounds)	Linear	KR	23	$2^{119.64}$	2^{64}	–	[5]

the obvious discrepancy in the requirements the unsuitability can be concluded. Each of the researched algorithms has specific advantages and disadvantages. For example, the implementation of KTANTAN32 is very compact and LED can be adjusted to the security level. However, the first step by choosing an encryption algorithm should always be the selection from algorithms which were developed for the own environment. All research algorithms have specific vulnerabilities for which already specific attacks exist. In each described attack the algorithms had been aware weakened or other specific conditions (physical access on hardware on specific points) had been created due the attack had been made possible. In many cases this factors can be fully excluded by the type of implementation. In this case there is no reason against a further usage of the researched algorithms. However, it should be always used the newest version of the respective algorithm because often with each new version vulnerabilities had been eliminated or the efficiency increased. This is as the AES example shows partly also for new implementations of existing algorithms. It is important and useful to develop and use different encryption algorithms for different use cases. Based on the current state of technology there is no universal algorithm which can be fully recommended for each use case.

References

1. V. R. ANDEM. A cryptanalysis of the tiny encryption algorithm. University of Alabama.
2. B. Andrews. Tiny encryption algorithm (tea) cryptography 4005.705.01 graduate team acd final report.
3. A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe. Present: An ultra-lightweight block cipher. In *Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '07, pages 450–466, Berlin, Heidelberg, 2007. Springer-Verlag.
4. C. Cannière, O. Dunkelman, and M. Knežević. Katan and ktantan – a family of small and efficient hardware-oriented block ciphers. In *Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems*, CHES '09, pages 272–288, Berlin, Heidelberg, 2009. Springer-Verlag.
5. J. Chen, M. Wang, and B. Preneel. Impossible differential cryptanalysis of the lightweight block ciphers tea, xtea and hight. In *AFRICACRYPT'12*, pages 117–137, 2012.
6. P. Derbez, P. Fouque, and J. Jean. Improved key recovery attacks on reduced-round AES in the single-key setting. In *Advances in Cryptology - EUROCRYPT 2013, 32nd Annual International Conference on the Theory and Applications of Cryptographic Techniques, Athens, Greece, May 26-30, 2013. Proceedings*, pages 371–387, 2013.
7. T. Eisenbarth, Z. Gong, T. Güneysu, S. Heyse, S. Indesteege, S. Kerckhof, F. Koeune, T. Nad, T. Plos, F. Regazzoni, F.-X. Standaert, and L. van Oldeneel tot Oldenzeel. Compact implementation and performance evaluation of block ciphers in attiny devices. In *Proceedings of the 5th International Conference on Cryptology in Africa*, AFRICACRYPT'12, pages 172–187, Berlin, Heidelberg, 2012. Springer-Verlag.
8. T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel. A survey of lightweight-cryptography implementations. *IEEE Des. Test*, 24(6):522–533, Nov. 2007.
9. M. Feldhofer, J. Wolkerstorfer, and V. Rijmen. Aes implementation on a grain of sand. *IEE Proceedings - Information Security*, 152:13–20(7), October 2005.
10. V. Grosso, C. Boura, B. Gérard, and F.-X. Standaert. A note on the empirical evaluation of security margins against algebraic attacks (with application to low cost-ciphers led and piccolo). In *33rd WIC Symposium on Information Theory in the Benelux*, 2012.
11. J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw. The led block cipher. In *Proceedings of the 13th International Conference on Cryptographic Hardware and Embedded Systems*, CHES'11, pages 326–341, Berlin, Heidelberg, 2011. Springer-Verlag.
12. P. Hamalainen, T. Alho, M. Hannikainen, and T. D. Hamalainen. Design and implementation of low-area and low-power aes encryption hardware core. In *Proceedings of the 9th EUROMICRO Conference on Digital System Design*, DSD '06, pages 577–583, Washington, DC, USA, 2006. IEEE Computer Society.
13. T. Isobe and K. Shibutani. Security analysis of the lightweight block ciphers xtea, led and piccolo. In *Proceedings of the 17th Australasian Conference on Information Security and Privacy*, ACISP'12, pages 71–86, Berlin, Heidelberg, 2012. Springer-Verlag.

14. P. Israsena. Design and implementation of low power hardware encryption for low cost secure rfid using tea.
15. P. Israsena and S. Wongnamkum. Hardware implementation of a tea-based lightweight encryption for rfid security.
16. P. Jovanovic, M. Kreuzer, I. Polian, and U. Passau. An algebraic fault attack on the led block cipher.
17. Y. Ko, S. Hong, W. Lee, S. Lee, and J.-S. Kang. Related key differential attacks on 27 rounds of xtea and full-round gost. In B. Roy and W. Meier, editors, *Fast Software Encryption*, volume 3017 of *Lecture Notes in Computer Science*, pages 299–316. Springer Berlin Heidelberg, 2004.
18. J. Lu. Related-key rectangle attack on 36 rounds of the xtea block cipher. *Int. J. Inf. Secur.*, 8(1):1–11, Jan. 2009.
19. F. Mendel, V. Rijmen, D. Toz, and K. Varici. Differential analysis of the led block cipher. *IACR Cryptology ePrint Archive*, 2012:544, 2012. informal publication.
20. A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang. Pushing the limits: A very compact and a threshold implementation of aes. In *Proceedings of the 30th Annual International Conference on Theory and Applications of Cryptographic Techniques: Advances in Cryptology*, EUROCRYPT'11, pages 69–88, Berlin, Heidelberg, 2011. Springer-Verlag.
21. I. Nikolic, L. Wang, and S. Wu. Cryptanalysis of round-reduced LED. *IACR Cryptology ePrint Archive*, 2015:429, 2015.
22. G. Sekar. Meet-in-the-middle attacks on reduced-round xtea. Belgium.
23. D. J. Wheeler. Correction to xtea. Cambridge University England.
24. D. J. Wheeler. Tea, atin yencryption algorithm. Cambridge University England.
25. Wikipedia. Nothing up my sleeve number. visited 28-July-2015.
26. Wikipedia. Kryptologische hashfunktion. 2015. visited 21-November-2015.
27. Wikipedia. Tiny encryption algorithm. 2015. visited 28-Aug-2015.
28. Wikipedia. Xtea. 2015. visited 15-July-2015.
29. D. Williams. The tiny encryption algorithm (tea). Columbus State University.
30. G. Wirtschaftslexikon. Internet der dinge. 2015. visited 28-July-2015.
31. E. Yarrkov. Cryptanalysis of xxtea.
32. X. Zhao, S. Guo, F. Zhang, T. Wang, Z. Shi, and K. Ji. Algebraic differential fault attacks on led using a single fault injection, 2012.