Information Processing Letters 115 (2015) 359–367

Contents lists available at ScienceDirect

Information Processing Letters

www.elsevier.com/locate/ipl

AKF: A key alternating Feistel scheme for lightweight cipher designs

F. Karakoç [a,b,**], H. Demirci [a,*], A.E. Harmancı [b]

[a] TÜBİTAK – BİLGEM – UEKAE, PK 74, 41470, Gebze, Kocaeli, Turkey
[b] Istanbul Technical University, Department of Computer Engineering, Faculty of Computer and Informatics, 34469, Maslak, Istanbul, Turkey

ARTICLE INFO

Article history:
Received 25 May 2014
Received in revised form 2 October 2014
Accepted 10 October 2014
Available online 22 October 2014
Communicated by Marc Fischlin

Keywords:
Cryptography
Lightweight block cipher
Key alternating cipher
Wireless sensor nodes
Feistel

ABSTRACT

In the classical Feistel structure the usage of alternating keys makes the cipher insecure against the related key attacks. In this work, we propose a new block cipher scheme, AKF, based on a Feistel structure with alternating keys but resistant against related key attacks. AKF leads constructions of lightweight block ciphers suitable for resource restricted devices such as RFID tags and wireless sensor nodes.

Using AKF we also present a software oriented lightweight block cipher, ITUBEE, especially suitable for wireless sensor nodes. We show that ITUBEE has a better performance than most of the ciphers which were compared in a recent work.

1. Introduction

Ubiquitous computing has been getting prominent because of the increase in daily life utilization. In this type of applications resource constrained devices such as RFID tags and sensor nodes are deployed. To meet the security and privacy issues in the applications cryptographic primitives which require less resources have been proposed under the topic lightweight cryptography. A lightweight block cipher is a main primitive in cryptographic requirements for ubiquitous computation. The need for lightweight block ciphers has triggered a lot of cipher constructions: PRESENT [1], PRINTCIPHER [2], LED [3], Prince [4], HIGHT [5], KLEIN [6], DESXL [7], KATAN [8], mCrypton [9], SEA [10], TEA [11] and LBlock [12].

Some of the proposed ciphers include novel ideas and challenging rationales while some of them have standard structures. One of the challenging rationale is the lack of key schedules such as done in PRINTCIPHER which can be included in Type 1A category introduced in [13]. Another way is to use key alternating cipher designs addressed in [14]. Key alternating ciphers are based on the Even–Mansour Scheme proposed in [15]. The definition of the scheme is $E_{F,k_1,k_2} = F(P \oplus k_1) \oplus k_2$ where F is a publicly known permutation over n-bit strings, k_1 and k_2 are n-bit secret keys and P is a plaintext. Nowadays there has been a lot of work on analysis of this scheme and iterated Even–Mansour scheme (called also as key alternating cipher) which is depicted in Fig. 1 [16–19] and there is a recent work presented at FSE 2014 which is about on security analysis of key alternating Feistel ciphers (KAF) [20].

While some ciphers based on iterated Even–Mansour scheme have been proposed such as LED and Prince, to the best of our knowledge there is no cipher based on key alternating Feistel scheme. GOST can be given as an example cipher based on a Feistel structure and a key schedule analogous to the key alternating cipher's schedule [21].

* Corresponding author. Tel.: +90 262 648 1737; fax: +90 262 648 1100.
** Principal corresponding author. Tel.: +90 262 648 1789; fax: +90 262 648 1100.
E-mail addresses: ferhat.karakoc@tubitak.gov.tr (F. Karakoç), huseyin.demirci@tubitak.gov.tr (H. Demirci), harmanci@itu.edu.tr (A.E. Harmancı).

http://dx.doi.org/10.1016/j.ipl.2014.10.010
0020-0190/

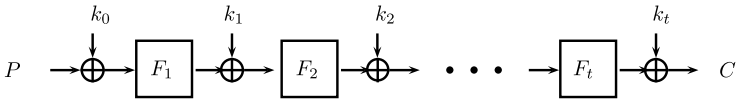

Fig. 1. A t-round key alternating cipher.

Usage of key alternating Feistel scheme gives some advances over performances of ciphers. Because of the Feistel structure same program code can be used both for encryption and decryption operations which reduces the memory usage. Also with the usage of no key schedule the memory and time requirements can be decreased. However, the nonexistence of a key schedule or the usage of alternating keys in a Feistel structure makes the cipher insecure against related key attacks [22]. In this study, we construct a block cipher scheme, AKF, using a Feistel structure with alternating keys in such a way that the security of our newly proposed scheme may not be altered. AKF is the first scheme which includes key alternating and Feistel structure providing security against related key attacks while key alternating Feistel ciphers are generally vulnerable to related key attacks as in the case of GOST [22].

In addition, using this scheme we reintroduce a new software oriented lightweight block cipher, ITUBEE. This cipher is especially designed for microcontroller based resource constrained devices having a limited battery power such as wireless sensor nodes. We have emulated the execution of ITUBEE on the Atmel ATtiny45 8-bit microcontroller using Atmel Studio 6 and evaluated the energy consumption and memory usage of the cipher. The results show that ITUBEE consumes less energy than most of the ciphers whose performance results were given in a recent work [23]. Also, less memory requirement of ITUBEE is noticeable.

The paper is structured as follows. In Section 2, we introduce our novel cipher scheme AKF and analyze the security including the related key attacks. Then in Section 3, we give the definition of ITUBEE with design rationale, security analysis, and performance results. Section 4 concludes the study.

2. A key alternating Feistel scheme (AKF)

2.1. Notation

Throughout the paper, we have used the following notation. P_L and P_R (C_L and C_R) denote the left and right halves of plaintext P (ciphertext C) respectively. We have used $\|$ to show the concatenation operation of two bit strings. i-th round key and round constant has been denoted by RK_i and RC_i respectively. k_i represents the parts of master key K where $K = k_0\|k_1\|...\|k_{t-1}$ and t is the number of key parts.

2.2. Definition

$\text{AKF}^{r,t}_{F_1,...,F_{2r}}$ is an r-round block cipher based on a Feistel structure with $2n$-bit block size and t n-bit keys $(k_0, ..., k_{t-1})$ where $F_1, ..., F_{2r}$ are publicly known permutations over n-bit strings. The encryption process is given in Algorithm 1 and pictured in Fig. 2.

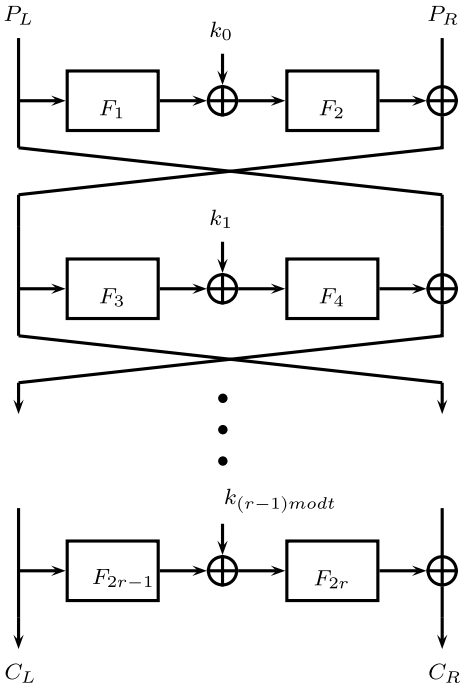

Fig. 2. $\text{AKF}^{r,t}_{F_1,...,F_{2r}}$ encryption algorithm.

Algorithm 1 $\text{AKF}^{r,t}_{F_1,...,F_{2r}}$ encryption algorithm.

1: $X_1 = P_L$ and $X_0 = P_R$
2: **for** $i = 1$ to r **do**
3: $\quad X_{i+1} = X_{i-1} \oplus F_{2i}(k_{(i-1)\,mod\,t} \oplus F_{2i-1}(X_i))$
4: **end for**
5: $C_L = X_r$ and $C_R = X_{r+1}$

Algorithm 2 $\text{AKF}^{r,t}_{F}$ encryption and decryption algorithm.

1: I, $(k_0, k_1, ..., k_{t-1})$, $(p_1, p_2, ..., p_{2r})$ are input parameters.
2: $X_1\|X_0 = I$.
3: **for** $i = 1$ to r **do**
4: $\quad X_{i+1} = X_{i-1} \oplus F_{2i}(k_{(i-1)\,mod\,t} \oplus F_{2i-1}(X_i))$
5: **end for**
6: $O = X_r\|X_{r+1}$.
7: Return O.

To use the same program code for encryption and decryption operations, each permutation F_i can be derived from a permutation F by using a parameter p_i such as $F_i = F(p_i)$ where p_i can be regarded as a round constant. In this case Algorithm 2 can be called for encryption and decryption operations with the parameters $\{P, (k_0, k_1, ..., k_{(t-1)}), (p_1, p_2, ..., p_{2r})\}$ and $\{C, (k_{(r-1)\,mod\,t}, k_{(r-2)\,mod\,t}, ..., k_{(r-t)\,mod\,t}), (p_{2r-1}, p_{2r}, ..., p_1, p_2)\}$ respectively.

For an encryption operation the input parameter I of the algorithm is a plaintext and the output parameter O is

the corresponding ciphertext while for a decryption operation I is a ciphertext and O is the corresponding plaintext.

2.3. Security analysis

In this section, we give security analysis of AKF scheme. We mainly focus on the related key attacks since Feistel cipher without key schedules are vulnerable against this type of attacks. We show that using AKF scheme it is easy to construct ciphers secure against related key attacks considering the analysis results given in Section 2.3.1. Also, we analyze the security of AKF with 2 keys, 3 keys and 4 keys because the common key sizes of the ciphers are n, $3n/2$ and $2n$ where n is the block size. We propose attacks on 5-round AKF with 2 alternating keys, 5- and 6-round AKF with 3 keys and 7-round AKF with 4 keys. In addition, we propose a general attack on $(2t-1)$-round AKF with t keys concluding that AKF with t keys is secure when the number of rounds is at least $2t$ where $t \geq 4$. We choose t as 4 or more because for $t = 1$ and $t = 2$ there are basic attacks and for $t = 3$ there is an impossible attack on 6 rounds. In the analysis we consider the permutations as black boxes.

As a conclusion, we observed that the proposed scheme is secure considering the required number of rounds. We claim that the 2, 3, 4 and t key versions of the scheme are secure when the minimum number of rounds are 6, 7, 8 and $2t$ respectively. Therefore, AKF construction satisfies enough security margin using relatively less rounds.

2.3.1. Related key attacks

In related key attack models, the attacker can have the plaintext/ciphertext pairs under the secret key K and under the key $R(K)$ where R is a relation chosen by the attacker. One of the mostly used related key attack techniques is the related key differential attack. In this attack, generally $R(K)$ is chosen as $K \oplus \Delta K$ where ΔK is a difference defined by the attacker. With the following propositions we show that AKF is resistant against related key differential attacks.

Proposition 1. *In a related key differential attack scenario, if the two keys have a difference only on the key part k_i then at least one permutation in the i-th round will be active.*

Proof. There are two cases: the left half of the round is active or passive. If the left half is active then it is obvious that the first permutation is active. In the other case the first permutation will be passive but after the first permutation the round key addition will make the input of the second permutation active because active and passive word addition will result an active word. Thus the second permutation will be active in this case. As a result if the round key is active then at least one permutation will be active on the round where active round key is used. □

Proposition 2. *If there exists a difference in the key then at least one permutation in consecutive t rounds will be active in the related key attack.*

Proof. When the key has a difference at least one key part out of $(k_0, k_1, ..., k_{t-1})$ will have a difference. In consecutive t rounds all key parts are used. Thus, at least in one round the active key part is used. Due to Proposition 1 there will exists at least one active permutation. □

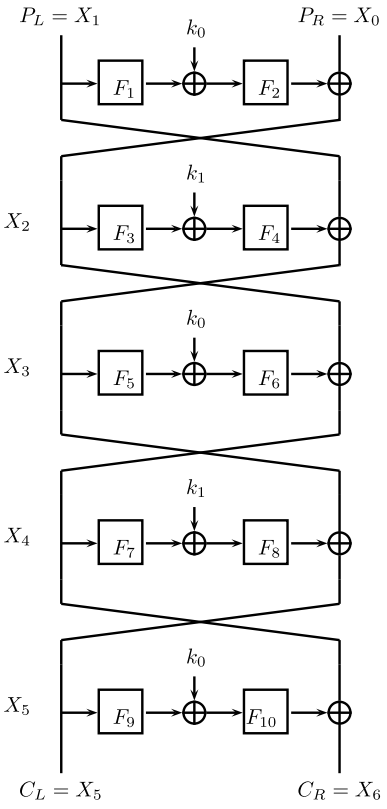

Fig. 3. AKF^{5,2}.

Let the maximum probability of a differential for the permutations be p. The minimum number of rounds on which a differential trail usable in a differential attack cannot be found is $\frac{t}{2^n \times p}$. Thus the minimum number of rounds could be decided considering this result.

2.3.2. An attack on 5-round AKF with 2 keys

In Algorithm 3 we present an attack on 5-round AKF with 2 keys pictured in Fig. 3. For the correct key the conditions in Step 6 and 8 in Algorithm 3 will be satisfied. For a wrong key the probability that the conditions will be satisfied is 2^{-3n}. Thus with a high probability we expect that only the correct key is returned by Algorithm 3. The attack algorithm uses only 2 plaintext–ciphertext pairs and finds the correct key in time 2^n. The memory requirement of the attack is negligible.

2.3.3. An attack on 5-round AKF with 3 keys

The round keys used in the first 2 rounds and last 2 rounds are k_0, k_1. k_2 is used only in the 3-rd round. Thus guessing k_0 and k_1 the key k_2 can be calculated easily using a plaintext–ciphertext pair. After this about $2^{3n-2n} = 2^n$ keys will remain. To find the correct key the remained keys can be tested using 1 plaintext–ciphertext pair. The time complexity of this attack is 2^{2n} encryption operations

Algorithm 3 An attack on $AKF^{5,2}$.

1: A plaintext–ciphertext pair (P, C) is given
2: **for** $k_0 = 0$ to $2^n - 1$ **do**
3: Perform the following computations:
- $X_2 \leftarrow X_0 \oplus F_2(k_0 \oplus F_1(X_1))$
- $X_4 \leftarrow X_6 \oplus F_{10}(k_0 \oplus F_9(X_5))$
- $X_3 \leftarrow F_5^{-1}(F_6^{-1}(X_2 \oplus X_4) \oplus k_0)$
- $k_1' \leftarrow F_3(X_2) \oplus F_4^{-1}(X_1 \oplus X_3)$
- $k_1'' \leftarrow F_7(X_4) \oplus F_8^{-1}(X_5 \oplus X_3)$

4: **if** $k_1' = k_1''$ **then**
5: $k_1 \leftarrow k_1'$
6: Test the key (k_0, k_1) using another (P, C) pair
7: If the test is ok then return the key as the correct key.
8: **end if**
9: **end for**

Algorithm 4 Impossible differential attack on $AKF^{6,3}$.

1: m plaintext pairs (P, P') and corresponding ciphertext pairs (C, C') are given where $P \oplus P' = (0, \Delta\alpha)$ and $C_L \oplus C_L' = \Delta\alpha$
2: **for** $k_2 = 0$ to $2^n - 1$ **do**
3: Compute X_5 and X_5' using C and C' as follows:
- $X_5 \leftarrow C_R \oplus F_{12}(k_2 \oplus F_{11}(C_L))$
- $X_5' \leftarrow C_R' \oplus F_{12}(k_2 \oplus F_{11}(C_L'))$

4: **if** $X_5 = X_5'$ **then**
5: Remove k_2 from candidate list where at the beginning the list contains all possible k_2 values.
6: **end if**
7: **end for**

because of the $2n$-bit key guess. The memory and data requirements are negligible.

2.3.4. An impossible differential attack on 6-round AKF with 3 keys

In this attack we use the following impossible differential characteristic [24] on 5-round Feistel cipher $(0, \Delta\alpha) \overset{5}{\nrightarrow} (0, \Delta\alpha)$. The attack is given in Algorithm 4.

The required number of plaintext pairs in Step 1 can be calculated as follows: For a wrong key the probability that the key is not eliminated using a plaintext pair is $(1-2^{-n})$ because of the condition in Step 4. Using m pairs this probability will be approximately $(1-2^{-n})^m$. There are 2^n possible keys so approximately $2^n \times (1-2^{-n})^m$ keys will remain. This should be less than or equal to 1 since we want to eliminate all wrong key candidates. We can find the following inequality from $2^n \times (1-2^{-n})^m$.

$$m \geq n \times \ln 2 \times 2^n$$

Using $n \times \ln 2 \times 2^n$ plaintext pairs we perform operations for 2^n possible values of k_2 so the time complexity of the attack algorithm is about $2^{2n} \times n \times \ln 2$.

2.3.5. An attack on 7-round AKF with 4 keys

This attack is similar to the attack given in Section 2.3.3. After guessing k_0, k_1 and k_2 the key k_3 used in the middle round can be calculated easily using a plaintext–ciphertext pair. The remained $2^{4n-2n} = 2^{2n}$ keys can be checked using 1 or 2 plaintext–ciphertext pairs. The time complexity of this attack is 2^{3n} encryption operations and the memory and data requirements are negligible. We cannot find an attack on 8 or more rounds.

Algorithm 5 ITUBEE encryption algorithm.

1: $X_1 \leftarrow P_L \oplus k_1$ and $X_0 \leftarrow P_R \oplus k_0$.
2: **for** $i = 1...20$ do **do**
3: **if** $i \in \{1, 3, ..., 19\}$ **then**
4: $RK_i \leftarrow k_0$
5: **else**
6: $RK_i \leftarrow k_1$
7: **end if**
8: $X_{i+1} \leftarrow X_{i-1} \oplus F(L(RK_i \oplus RC_i \oplus F(X_i)))$ where 16-bit round constant RC_i is added to the rightmost 16 bits
9: **end for**
10: $C_L \leftarrow X_{20} \oplus k_0$ and $C_R \leftarrow X_{21} \oplus k_1$

2.3.6. Attacks on $(2t-1)$-round AKF with t keys

For $t \geq 3$ we propose the following attack on $AKF^{(2t-1),t}$. The attack works as follows: Guess the keys $k_0, k_1, ..., k_{t-2}$ and calculate the key k_{t-1} which is used in the middle round. Then check the remained 2^{tn-2n} keys using $\frac{tn-2n}{2n}$ plaintext–ciphertext pairs.

For more than $(2t-1)$ rounds we could not find any attack. We claim that AKF with t keys are secure if the number of rounds is at least $2t$.

3. A software oriented lightweight block cipher (ITUBEE)

3.1. Definition

ITUBEE introduced in [25] is an example cipher of $AKF^{20,2}$ with 80-bit block size. In addition to the AKF scheme ITUBEE has also key whitening layers. The whitening keys at the top and bottom of the encryption process are $(k_1 \| k_0)$ and $(k_0 \| k_1)$, respectively. The permutations used on the left hand side in the rounds are same while the ones used on the right hand side differ from each other because of the round constant addition.

Algorithm 5 presents and Fig. 4 illustrates the encryption process of the cipher.

The functions F and L used in Algorithm 5 are defined as:

- $S(a\|b\|c\|d\|e) = s[a]\|s[b]\|s[c]\|s[d]\|s[e]$ where a, b, c, d, e are 8-bit variables and s is the AES S-box [26].
- $L(a\|b\|c\|d\|e) = (e \oplus a \oplus b)\|(a \oplus b \oplus c)\|(b \oplus c \oplus d)\|(c \oplus d \oplus e)\|(d \oplus e \oplus a)$.
- $F(X) = S(L(S(X)))$.

The constants given in Table 1 are selected as round constants (RC_i).

The decryption process is very similar to the encryption process. The only differences are on the order of the keys and round constants. In the decryption operation, the master key is $(k_1 \| k_0)$ and the round constants are used in reversed order.

We claim that the security level of ITUBEE is 80-bit. Also, this security level is valid for the related key attack model.

3.2. Design rationale

ITUBEE is especially designed for wireless sensor nodes having limited battery power. The energy consumption is directly related to the number of instructions in an encryption operation. Thus, constructing the permutations in AKF

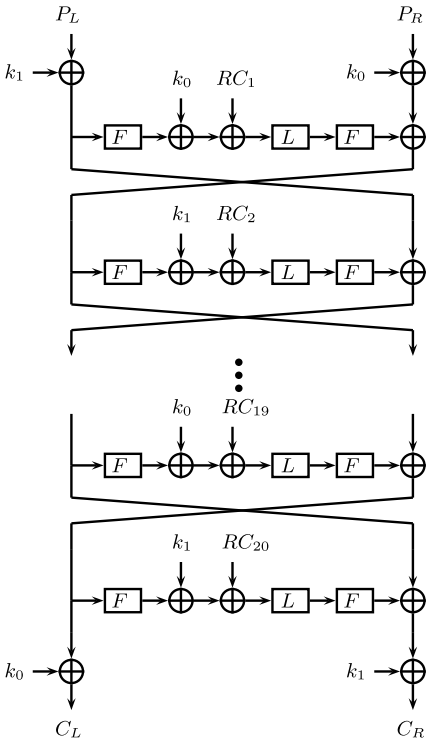

Fig. 4. ITUBEE encryption algorithm.

Table 1
Round constants of ITUBEE.

i	RC_i	i	RC_i	i	RC_i	i	RC_i
1	0x1428	6	0x0f23	11	0x0a1e	16	0x0519
2	0x1327	7	0x0e22	12	0x091d	17	0x0418
3	0x1226	8	0x0d21	13	0x081c	18	0x0317
4	0x1125	9	0x0c20	14	0x071b	19	0x0216
5	0x1024	10	0x0b1f	15	0x061a	20	0x0115

scheme we tried to use minimum number of instructions. Also we paid attention to minimize the memory usage in the design of the cipher.

In a software oriented platform the easiest way for a nonlinear operation is the usage of a substitution box because a good confusion can be satisfied with a few number of operations. Since we target 8-bit platforms we preferred to use an 8-bit S-box. Also, we used a cellular automaton which can be implemented with just 15 XOR operations for the diffusion layer.

We used the same F function in the rounds on the left hand side to make the construction minimal. For permutations on the right hand side we also used the same F function including a round constant addition to prevent the self-similarity attacks such as reflection [27], slide [28], and slidex [16]. To reduce the number of operations we used 16-bit constants instead of the whole block size done in [2]. We didn't prefer to use 8-bit constants since it may cause the leakage of information as follows:

Proposition 3. *If the inputs of the F function have the pattern* (a, b, b, a, c) *where* a, b, c *are 8-bit values then the output will also have the same pattern.*

Proof. The first S-box layer saves the pattern. The output of the linear layer will be $((s[c] \oplus s[a] \oplus s[b])\|(s[a] \oplus s[b] \oplus s[b])\|(s[b] \oplus s[b] \oplus s[a])\|(s[b] \oplus s[a] \oplus s[c])\|(s[a] \oplus s[c] \oplus s[a])) = ((s[c] \oplus s[a] \oplus s[b])\|s[a]\|s[a]\|s([b] \oplus s[a] \oplus s[c])\|s[c])$ which have the same pattern. As a result, F saves the pattern. □

Proposition 4. *If* ITUBEE *uses 1-byte round constants added on the rightmost byte and the input and the key have the pattern* $(a\|b\|b\|a\|c\|d\|e\|e\|d\|f)$ *then the corresponding ciphertext will have the same pattern also.*

Proof. If the input and the key have the pattern

$$(a\|b\|b\|a\|c\|d\|e\|e\|d\|f)$$

then P_L, P_R, K_L, K_R have the same pattern given in Proposition 3. The key additions in the algorithm will not change the pattern. In addition, F, L and S-box layers will not change the pattern where the reason is given in Proposition 3. The round constant only changes the rightmost byte which does not effect the pattern. As a conclusion, C_L and C_R will have the same pattern. Note that the number of rounds does not effect the result. □

3.3. Security analysis

3.3.1. Differential and linear cryptanalysis

Two of the mostly used cryptanalysis techniques are the differential cryptanalysis [29] and the linear cryptanalysis [30] which are a kind of statistical analysis. In these techniques, ciphers are modeled as linear algorithms with a probability replacing the nonlinear operations in ciphers with linear operations. The probability of modeling the cipher as a linear algorithm depends on the number of replacement of nonlinear operations with the probabilities of the replacements. The effect of these cryptanalysis techniques are strongly correlated with this probability. Thus, maximizing the probability by minimizing the number of replacement of nonlinear operations is a major work in these cryptanalysis techniques.

The S-box is the only nonlinear operation in ITUBEE. Thus, finding linear and differential trails which have less number of active S-boxes is a main work in linear and differential cryptanalysis, respectively. The number of active S-boxes with linear and differential probabilities gives the security of the cipher against the linear and differential cryptanalysis.

We give the following propositions about the number of active S-boxes and their probabilities.

Proposition 5. *The minimum number of active S-boxes both in a differential trail and in a linear trail for 3 consecutive rounds is 16.*

Proof. It is trivial to calculate the branch number of the F function as 4 by searching all possible cases. That gives at least 4 active S-boxes. For one round if the left half of the input is active then we have 2 active F functions. Lets denote a Feistel round as $(L^{i+1}, R^{i+1}) = (G(L^i) \oplus R^i, L^i)$ where $L^i\|R^i$ and $L^{i+1}\|R^{i+1}$ are input and outputs of the round

respectively and G is a function. If G is one-to-one then at least 2 G functions are active for 3 consecutive rounds. For ITUBEE the G function consists of 2 F and 1 L functions with key addition and constant addition and therefore G function is one-to-one. Thus for 3 consecutive rounds we have at least 2 G functions consisting of 4 F functions and so 16 active S-boxes. □

We found the following differential and linear trails for G function where the number of active S-boxes is 8. By using these trails 3-round trails can be constructed easily where one out of three G functions is passive and the total number of active S-boxes is 16. The differential and linear trails are

- $(\Delta a\|0\|0\|0\|0) \xrightarrow{S} (\Delta b\|0\|0\|0\|0) \xrightarrow{L} (\Delta b\|\Delta b\|0\|0\|\Delta b) \xrightarrow{S} (\Delta c\|\Delta c\|0\|0\|\Delta c) \xrightarrow{L} (\Delta c\|0\|\Delta c\|\Delta c\|0) \xrightarrow{S} (\Delta d\|0\|\Delta d\|\Delta d\|0) \xrightarrow{L} (\Delta d\|0\|0\|0\|0) \xrightarrow{S} (\Delta a\|0\|0\|0\|0)$

and

- $(\Gamma a\|0\|0\|0\|0) \xrightarrow{S} (\Gamma b\|0\|0\|0\|0) \xrightarrow{L} (\Gamma b\|0\|\Gamma b\|\Gamma b\|0) \xrightarrow{S} (\Gamma c\|0\|\Gamma c\|\Gamma c\|0) \xrightarrow{L} (\Gamma c\|\Gamma c\|0\|0\|\Gamma c) \xrightarrow{S} (\Gamma d\|\Gamma d\|0\|0\|\Gamma d) \xrightarrow{L} (\Gamma d\|0\|0\|0\|0) \xrightarrow{S} (\Gamma a\|0\|0\|0\|0)$

respectively.

ITUBEE uses AES S-box whose highest probability for one input–output difference is 2^{-6} and the best bias for an input–output mask is $\mp 2^{-4}$. Thus, the highest probability for a differential trail on consecutive 3 rounds is $(2^{-6})^{16} = 2^{-96}$. Similarly, the best linear bias for 3-round linear trail is $2^{15} \times (\mp 2^{-4})^{16} = \mp 2^{-49}$. According to these results it seems that these trails are not usable in differential and linear attacks.

However, to prove a cipher's resistance against differential and linear attacks computation of active number of S-boxes is not enough because of the multiple differentials and linear hulls. For linear hulls we left the theoretical computations to the third party cryptanalysis and we experimentally computed the bias of a linear trail for F function where 4 S-boxes are active to see how the bias changes. In the case of multiple differentials we gave some theoretical and experimental results.

In a linear trail for F function the minimum number of active S-boxes is 4 as stated above. An example of such a trail is $(\Gamma a\|0\|0\|0\|0) \xrightarrow{S} (\Gamma b\|0\|0\|0\|0) \xrightarrow{L} (\Gamma b\|0\|\Gamma b\|\Gamma b\|0) \xrightarrow{S} (\Gamma c\|0\|\Gamma c\|\Gamma c\|0)$. In this trail when the input–output masks of the S-boxes which have the best bias ($\mp 2^{-4}$) are chosen, the bias of the linear characteristic becomes $2^3 \times (\mp 2^{-4})^4 = \mp 2^{-13}$. The masks can be chosen as $\Gamma a = 0x10$, $\Gamma b = 0x65$ and $\Gamma c = 0xa3$. We tested the bias of this linear characteristic using 2^{28} randomly chosen inputs and computed the bias as about 2^{-11}.

With the following proposition we give an upper bound for the probability of a differential on F function.

Proposition 6. *The maximum probability of a differential for F function is less than* 2^{-17}.

Proof. There are 3 different cases which lead to 4 active S-boxes. Let Δa, Δb, Δc, Δd, Δe denote 8-bit differences and 0 denotes 8 bits which don't have any difference. Then these cases can be written as:

1. $(\Delta a\|0\|0\|0\|0) \xrightarrow{S} (\Delta b\|0\|0\|0\|0) \xrightarrow{L} (\Delta b\|\Delta b\|0\|0\|\Delta b) \xrightarrow{S} (\Delta c\|\Delta d\|0\|0\|\Delta e)$.
   Note that changing the place of Δa is not important because of the symmetric structure of the L layer.
2. $(\Delta a\|\Delta b\|0\|0\|0) \xrightarrow{S} (\Delta c\|\Delta c\|0\|0\|0) \xrightarrow{L} (0\|0\|\Delta c\|0\|\Delta c) \xrightarrow{S} (0\|0\|\Delta d\|0\|\Delta e)$.
3. $(\Delta a\|\Delta b\|0\|\Delta c\|0) \xrightarrow{S} (\Delta d\|\Delta d\|0\|\Delta d\|0) \xrightarrow{L} (0\|0\|0\|\Delta d\|0) \xrightarrow{S} (0\|0\|0\|\Delta e\|0)$.

The probabilities of the cases can be calculated as:

1. $\sum_{\Delta b} Pr(\Delta a \xrightarrow{S} \Delta b) \times Pr(\Delta b \xrightarrow{S} \Delta c) \times Pr(\Delta b \xrightarrow{S} \Delta d) \times Pr(\Delta b \xrightarrow{S} \Delta e) \leq (2^{-6})^4 \times 2^7 = 2^{-17}$ because the maximum probability of a differential for the AES S-box is 2^{-6} and there are 127 possible Δb for a given difference Δa.
2. $\sum_{\Delta c} Pr(\Delta a \xrightarrow{S} \Delta c) \times Pr(\Delta b \xrightarrow{S} \Delta c) \times Pr(\Delta c \xrightarrow{S} \Delta d) \times Pr(\Delta c \xrightarrow{S} \Delta e) \leq (2^{-6})^4 \times 2^7 = 2^{-17}$.
3. $\sum_{\Delta d} Pr(\Delta a \xrightarrow{S} \Delta d) \times Pr(\Delta b \xrightarrow{S} \Delta d) \times Pr(\Delta c \xrightarrow{S} \Delta d) \times Pr(\Delta d \xrightarrow{S} \Delta e) \leq (2^{-6})^4 \times 2^7 = 2^{-17}$.

As seen for all the cases which leads to 4 active S-boxes the differential probabilities are less than 2^{-17}. For the other cases it is trivial to see that the probabilities of the differentials will be less than 2^{-17}. □

We observed that the upper bound given in Proposition 6 is not tight because in the proof of the proposition the probabilities of input–output differences for the S-box were chosen as 2^{-6} while the probability is 2^{-7} for most of the input–output differences. By using the difference distribution table of the S-box we computed the probabilities of the following 3 trails searching all the possible values of Δa, Δb and Δc differences. Note that these 3 trails covers all possible trails activating 4 S-boxes.

1. $(\Delta a\|0\|0\|0\|0) \xrightarrow{S} (\Delta b\|0\|0\|0\|0) \xrightarrow{L} (\Delta b\|\Delta b\|0\|0\|\Delta b) \xrightarrow{S} (\Delta c\|\Delta c\|0\|0\|\Delta c)$.
2. $(\Delta a\|\Delta a\|0\|0\|0) \xrightarrow{S} (\Delta b\|\Delta b\|0\|0\|0) \xrightarrow{L} (0\|0\|\Delta b\|0\|\Delta b) \xrightarrow{S} (0\|0\|\Delta c\|0\|\Delta c)$.
3. $(\Delta a\|\Delta a\|0\|\Delta a\|0) \xrightarrow{S} (\Delta b\|\Delta b\|0\|\Delta b\|0) \xrightarrow{L} (0\|0\|0\|\Delta b\|0) \xrightarrow{S} (0\|0\|0\|\Delta c\|0)$.

We found that the best probability is $2^{-21.51}$ (when $\Delta a = 0x75$ and $\Delta c = 0xd8$ we got the probability). We tested the multiple differential effect using 2^{28} randomly chosen inputs and setting the input–output differences for F as $(0x75\|0\|0\|0\|0)$–$(0xd8\|0xd8\|0\|0\|0xd8)$ and observed that the probability is close to the theoretical value $2^{-21.51}$.

To analyze the multiple differential effect on G function we also computed the highest probability for the trails

Table 2
Performance results of some lightweight ciphers.

Cipher	Block size [bits]	Key size [bits]	Memory code/RAM [bytes]	Clock cycles per one enc.	Clock cycles per one byte	Cycle × Memory
AES [23]	128	128	1659/33	4557	284	479 676
DESXL [23]	64	184	820/48	84 602	10 575	9 179 100
HIGHT [23]	64	128	402/32	19 503	2437	1 057 658
IDEA [23]	64	128	836/232	≈ 8250	1031	1 101 108
KASUMI [23]	64	128	1264/24	11 939	1492	1 921 696
KATAN [23]	64	80	338/18	72 063	9007	3 206 492
KLEIN [23]	64	80	1268/18	6095	761	978 646
mCrypton [23]	64	96	1076/28	16 457	2057	2 270 928
NOEKEON [23]	128	128	364/32	23 517	1469	581 724
PRESENT [23]	64	80	1000/18	11 342	1417	1 442 506
SEA [23]	96	96	426/24	41 604	3467	1 560 150
TEA [23]	64	128	648/24	7408	926	622 272
LBlock [12]	64	80	not given	3955	494	–
ITUBEE cycle opt.	80	80	716/20	2607	261	192 096
ITUBEE memory opt.	80	80	586/20	2937	294	178 164
AES	128	128	1028/41	3818	238	254 422

on G function given above and found that the highest probability is about $(2^{-35.86})$ (when we take $\Delta a = 0x08$ we obtained this probability). According to this result for a 5-round differential the maximum probability is about $(2^{-35.86})^3 = 2^{107.58}$ which cannot be used in a differential attack.

3.3.2. Meet-in-the-middle type attacks

In the basic meet-in-the-middle (MITM) attacks the cipher is divided into two parts. The effects of the key bits are separated from each part so that a check point is derived in the middle. In the case of ITUBEE, in consecutive 3 rounds all key bits are used and each output bits of 3 rounds is affected by each bit of the key. Thus, more than 6-round ITUBEE because of the confusion property cannot be attacked using the basic MITM type techniques.

In recent years, there have been new attack techniques based on MITM such as multidimensional meet-in-the-middle attack [31], the biclique attack [32]. For the biclique cryptanalysis of the cipher, let the whitening keys do not exist. A biclique can be constructed on at most 2 rounds and the number of F functions computed for the whole key is about 32 for the meet-in-the-middle step of the attack. Therefore, a biclique attack complexity is about $\frac{32\times 2^{80}}{40} \approx 2^{79.678}$. As a result, biclique type attacks cannot significantly reduce the security level of the cipher.

The multidimensional meet-in-the-middle attack is only applied on a cipher if the key length of the cipher is bigger than the block size. The block size and key length of ITUBEE are equal. Thus this attack cannot affect the security level of the cipher.

3.3.3. Related key differential attacks

By Proposition 2, the number of active F functions for 10 rounds is 5. The maximum differential probability for one F is about 2^{-17}. Thus for 10 round the maximum probability is about $(2^{-17})^5 = 2^{-85}$. As a result there is not a 10-round differential which is usable in an attack.

3.3.4. Impossible differential attacks

Impossible differential attack [33] is one of the mostly applied attacks on lightweight block ciphers especially suitable for software platforms [34–39]. The maximum number of rounds on which an impossible characteristic exists is 5 because of the Feistel structure [24]. Thus this attack technique does not threat the security level of the cipher.

3.3.5. Self-similarity attacks

In self-similarity attacks [27,28,16] the similarities between round functions are used. The round functions of ITUBEE are very similar. However, the usage of round constants prevents the application of these type of attacks on the cipher.

3.4. Emulation results

We implemented ITUBEE encryption algorithm in assembly and emulated the execution of the implementation on the Atmel AVR ATtiny45 and ATmega128(L) microcontrollers using Atmel Studio 6. ATmega128(L) microcontrollers are included in Mica2 motes which are commonly used sensor nodes. These microcontrollers are RISC (Reduced Instruction Set Computing) based and have an 8-bit Harvard architecture. In the Harvard architecture the instruction and data memory are separated. ATtiny45 has an instruction and data memory of 4-kB Flash and 256-byte static RAM, respectively.

In the implementation of the cipher, 8-bit S-box is stored in the instruction memory. In the emulation we evaluated the memory usage of the cipher and the number of clock cycles for an encryption. The results are the same for the two microcontrollers and given in Table 2 with some other ciphers' results obtained by a recent work [23] where the implementations also were done on an ATtiny45 microcontroller. In Table 2 we also give the number of clock cycles per one byte. To have a fair comparison we implemented AES using the same methods we used for ITUBEE.

From the table it can be extracted that ITUBEE requires less clock cycles for an encryption than the other ciphers. According to [40,23] the number of clock cycles is strongly correlated to the energy consumption. Thus, combining these two results we conclude that ITUBEE consumes less energy than the other ciphers. Also, the table shows that the memory reduction of the proposed cipher is remarkable.

4. Conclusion

The usage of alternating keys facilitates to design lightweight ciphers because of the lack of a key schedule inducing less memory usage and fewer operation (energy) requirement. Also, using a Feistel structure gives the advantage of using the same program code for encryption and decryption processes. This reduces memory usage. However, using a Feistel structure with alternating keys could make the cipher insecure against related key attacks as in [22]. To palliate this weakness we have proposed a new design strategy and using this rationale we have presented a Feistel based key alternating scheme, AKF.

We also introduce a software oriented lightweight block cipher, ITUBEE, based on AKF. We have emulated the execution of the cipher on the 8-bit ATtiny45 and ATmega128(L) microcontrollers where ATmega128(L) is included in widely used Mica2 motes. The emulation results show that ITUBEE consumes less energy than most of the ciphers whose performance results were given in a recent work [23]. Also the reduction in memory requirement of the cipher is remarkable.

The utilization of AKF scheme enables us to develop a new block cipher (ITUBEE) whose memory usage and energy consumption is less than most of the existing block ciphers.

Appendix A. AES S-box

```
s[256] = {
63, 7C, 77, 7B, F2, 6B, 6F, C5,
30, 01, 67, 2B, FE, D7, AB, 76
CA, 82, C9, 7D, FA, 59, 47, F0,
AD, D4, A2, AF, 9C, A4, 72, C0
B7, FD, 93, 26, 36, 3F, F7, CC,
34, A5, E5, F1, 71, D8, 31, 15
04, C7, 23, C3, 18, 96, 05, 9A,
07, 12, 80, E2, EB, 27, B2, 75
09, 83, 2C, 1A, 1B, 6E, 5A, A0,
52, 3B, D6, B3, 29, E3, 2F, 84
53, D1, 00, ED, 20, FC, B1, 5B,
6A, CB, BE, 39, 4A, 4C, 58, CF
D0, EF, AA, FB, 43, 4D, 33, 85,
45, F9, 02, 7F, 50, 3C, 9F, A8
51, A3, 40, 8F, 92, 9D, 38, F5,
BC, B6, DA, 21, 10, FF, F3, D2
CD, 0C, 13, EC, 5F, 97, 44, 17,
C4, A7, 7E, 3D, 64, 5D, 19, 73
60, 81, 4F, DC, 22, 2A, 90, 88,
46, EE, B8, 14, DE, 5E, 0B, DB
E0, 32, 3A, 0A, 49, 06, 24, 5C,
C2, D3, AC, 62, 91, 95, E4, 79
E7, C8, 37, 6D, 8D, D5, 4E, A9,
6C, 56, F4, EA, 65, 7A, AE, 08
BA, 78, 25, 2E, 1C, A6, B4, C6,
E8, DD, 74, 1F, 4B, BD, 8B, 8A
70, 3E, B5, 66, 48, 03, F6, 0E,
61, 35, 57, B9, 86, C1, 1D, 9E
E1, F8, 98, 11, 69, D9, 8E, 94,
9B, 1E, 87, E9, CE, 55, 28, DF
8C, A1, 89, 0D, BF, E6, 42, 68,
41, 99, 2D, 0F, B0, 54, BB, 16
}
```

Appendix B. Test vectors for ITUBEE

Table B3
Test vector 1.

Plaintext	00000000000000000000
Key	00000000000000000000
Ciphertext	471330577984cbecf6c8

Table B4
Test vector 2.

Plaintext	01000000000000000000
Key	00000000000000000080
Ciphertext	761b8299b3f6a99f0838

Table B5
Test vector 3.

Plaintext	6925278951fbf3b25ccc
Key	c538bd9289822be43363
Ciphertext	c42e0f48cd5a87d0055f

References

[1] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin, C. Vikkelsoe, PRESENT: an ultra-lightweight block cipher, in: P. Paillier, I. Verbauwhede (Eds.), CHES, in: Lect. Notes Comput. Sci., vol. 4727, Springer, 2007, pp. 450–466.

[2] L.R. Knudsen, G. Leander, A. Poschmann, M.J.B. Robshaw, PRINTcipher: a block cipher for IC-printing, in: S. Mangard, F.-X. Standaert (Eds.), CHES, in: Lect. Notes Comput. Sci., vol. 6225, Springer, 2010, pp. 16–32.

[3] J. Guo, T. Peyrin, A. Poschmann, M.J.B. Robshaw, The LED block cipher, in: B. Preneel, T. Takagi (Eds.), CHES, in: Lect. Notes Comput. Sci., vol. 6917, Springer, 2011, pp. 326–341.

[4] J. Borghoff, A. Canteaut, T. Güneysu, E.B. Kavun, M. Knezevic, L.R. Knudsen, G. Leander, V. Nikov, C. Paar, C. Rechberger, P. Rombouts, S.S. Thomsen, T. Yalçin, PRINCE – a low-latency block cipher for pervasive computing applications – extended abstract, in: [41], pp. 208–225.

[5] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, B. Koo, C. Lee, D. Chang, J. Lee, K. Jeong, H. Kim, J. Kim, S. Chee, HIGHT: a new block cipher suitable for low-resource device, in: L. Goubin, M. Matsui (Eds.), CHES, in: Lect. Notes Comput. Sci., vol. 4249, Springer, 2006, pp. 46–59.

[6] Z. Gong, S. Nikova, Y.W. Law, KLEIN: a new family of lightweight block ciphers, in: A. Juels, C. Paar (Eds.), RFIDSec, in: Lect. Notes Comput. Sci., vol. 7055, Springer, 2011, pp. 1–18.

[7] G. Leander, C. Paar, A. Poschmann, K. Schramm, New lightweight DES variants, in: A. Biryukov (Ed.), FSE, in: Lect. Notes Comput. Sci., vol. 4593, Springer, 2007, pp. 196–210.

[8] C.D. Cannière, O. Dunkelman, M. Knezevic, KATAN and KTANTAN – a family of small and efficient hardware-oriented block ciphers, in: C. Clavier, K. Gaj (Eds.), CHES, in: Lect. Notes Comput. Sci., vol. 5747, Springer, 2009, pp. 272–288.

[9] C.H. Lim, T. Korkishko, mCrypton – a lightweight block cipher for security of low-cost RFID tags and sensors, in: J. Song, T. Kwon, M. Yung (Eds.), WISA, in: Lect. Notes Comput. Sci., vol. 3786, Springer, 2005, pp. 243–258.
[10] F.-X. Standaert, G. Piret, N. Gershenfeld, J.-J. Quisquater, SEA: a scalable encryption algorithm for small embedded applications, in: J. Domingo-Ferrer, J. Posegga, D. Schreckling (Eds.), CARDIS, in: Lect. Notes Comput. Sci., vol. 3928, Springer, 2006, pp. 222–236.
[11] D.J. Wheeler, R.M. Needham, TEA, a tiny encryption algorithm, in: B. Preneel (Ed.), FSE, in: Lect. Notes Comput. Sci., vol. 1008, Springer, 1994, pp. 363–366.
[12] W. Wu, L. Zhang, LBlock: a lightweight block cipher, in: J. Lopez, G. Tsudik (Eds.), ACNS, in: Lect. Notes Comput. Sci., vol. 6715, 2011, pp. 327–344.
[13] G. Carter, E. Dawson, L. Nielsen, Key schedules of iterative block ciphers, in: C. Boyd, E. Dawson (Eds.), ACISP, in: Lect. Notes Comput. Sci., vol. 1438, Springer, 1998, pp. 80–89.
[14] A. Bogdanov, L.R. Knudsen, G. Leander, F.-X. Standaert, J.P. Steinberger, E. Tischhauser, Key-alternating ciphers in a provable setting: encryption using a small number of public permutations – (extended abstract), in: [42], pp. 45–62.
[15] S. Even, Y. Mansour, A construction of a cipher from a single pseudorandom permutation, in: H. Imai, R.L. Rivest, T. Matsumoto (Eds.), ASIACRYPT, in: Lect. Notes Comput. Sci., vol. 739, Springer, 1991, pp. 210–224.
[16] O. Dunkelman, N. Keller, A. Shamir, Minimalism in cryptography: the Even–Mansour scheme revisited, in: [42], pp. 336–354.
[17] I. Dinur, O. Dunkelman, N. Keller, A. Shamir, Key recovery attacks on 3-round Even–Mansour, 8-step LED-128, and full AES2, in: K. Sako, P. Sarkar (Eds.), ASIACRYPT (1), in: Lect. Notes Comput. Sci., vol. 8269, Springer, 2013, pp. 337–356.
[18] R. Lampe, J. Patarin, Y. Seurin, An asymptotically tight security analysis of the iterated Even–Mansour cipher, in: [41], pp. 278–295.
[19] J.P. Steinberger, Improved security bounds for key-alternating ciphers via Hellinger distance, IACR Cryptol. ePrint Arch. 2012 (2012) 481.
[20] R. Lampe, Y. Seurin, Security analysis of key-alternating Feistel ciphers, IACR Cryptol. ePrint Arch. 2014 (2014) 151.
[21] I. Zabotin, G. Glazkov, V. Isaeva, Cryptographic protection for information processing systems. Cryptographic transformation algorithm, in: Government Standard of the USSR, GOST 28147-89, vol. 1989, 1989.
[22] Y. Ko, S. Hong, W. Lee, S. Lee, J.-S. Kang, Related key differential attacks on 27 rounds of XTEA and full-round GOST, in: B.K. Roy, W. Meier (Eds.), FSE, in: Lect. Notes Comput. Sci., vol. 3017, Springer, 2004, pp. 299–316.
[23] T. Eisenbarth, Z. Gong, T. Güneysu, S. Heyse, S. Indesteege, S. Kerckhof, F. Koeune, T. Nad, T. Plos, F. Regazzoni, F.-X. Standaert, L. van Oldeneel tot Oldenzeel, Compact implementation and performance evaluation of block ciphers in ATtiny devices, in: [43], pp. 172–187.
[24] L.R. Knudsen, The security of Feistel ciphers with six rounds or less, J. Cryptol. 15 (2002) 207–222.
[25] F. Karakoç, H. Demirci, A.E. Harmanci, ITUbee: a software oriented lightweight block cipher, in: G. Avoine, O. Kara (Eds.), LightSec, in: Lect. Notes Comput. Sci., vol. 8162, Springer, 2013, pp. 16–27.
[26] J. Daemen, V. Rijmen, The Design of Rijndael: AES – The Advanced Encryption Standard, Springer, 2002.
[27] O. Kara, Reflection cryptanalysis of some ciphers, in: D.R. Chowdhury, V. Rijmen, A. Das (Eds.), INDOCRYPT, in: Lect. Notes Comput. Sci., vol. 5365, Springer, 2008, pp. 294–307.
[28] A. Biryukov, D. Wagner, Slide attacks, in: L.R. Knudsen (Ed.), FSE, in: Lect. Notes Comput. Sci., vol. 1636, Springer, 1999, pp. 245–259.
[29] E. Biham, A. Shamir, Differential cryptanalysis of DES-like cryptosystems, in: A. Menezes, S.A. Vanstone (Eds.), CRYPTO, in: Lect. Notes Comput. Sci., vol. 537, Springer, 1990, pp. 2–21.
[30] M. Matsui, Linear cryptoanalysis method for DES cipher, in: T. Helleseth (Ed.), EUROCRYPT, in: Lect. Notes Comput. Sci., vol. 765, Springer, 1993, pp. 386–397.
[31] B. Zhu, G. Gong, Multidimensional meet-in-the-middle attack and its applications to KATAN32/48/64, Cryptology ePrint Archive, Report 2011/619, 2011.
[32] A. Bogdanov, D. Khovratovich, C. Rechberger, Biclique cryptanalysis of the full AES, in: D.H. Lee, X. Wang (Eds.), ASIACRYPT, in: Lect. Notes Comput. Sci., vol. 7073, Springer, 2011, pp. 344–371.
[33] E. Biham, A. Biryukov, A. Shamir, Cryptanalysis of Skipjack reduced to 31 rounds using impossible differentials, in: J. Stern (Ed.), EUROCRYPT, in: Lect. Notes Comput. Sci., vol. 1592, Springer, 1999, pp. 12–23.
[34] J. Chen, M. Wang, B. Preneel, Impossible differential cryptanalysis of the lightweight block ciphers TEA, XTEA and HIGHT, in: [43], pp. 117–137.
[35] F. Karakoç, H. Demirci, A.E. Harmanci, Impossible differential cryptanalysis of reduced-round LBlock, in: I.G. Askoxylakis, H.C. Pöhls, J. Posegga (Eds.), WISTP, in: Lect. Notes Comput. Sci., vol. 7322, Springer, 2012, pp. 179–188.
[36] Y. Liu, D. Gu, Z. Liu, W. Li, Impossible differential attacks on reduced-round LBlock, in: M.D. Ryan, B. Smyth, G. Wang (Eds.), ISPEC, in: Lect. Notes Comput. Sci., vol. 7232, Springer, 2012, pp. 97–108.
[37] Y. Liu, D. Gu, Z. Liu, W. Li, Improved results on impossible differential cryptanalysis of reduced-round Camellia-192/256, J. Syst. Softw. 85 (2012) 2451–2458.
[38] O. Özen, K. Varici, C. Tezcan, Çelebi Kocair, Lightweight block ciphers revisited: cryptanalysis of reduced round PRESENT and HIGHT, in: C. Boyd, J.M.G. Nieto (Eds.), ACISP, in: Lect. Notes Comput. Sci., vol. 5594, Springer, 2009, pp. 90–107.
[39] W. Wu, L. Zhang, W. Zhang, Improved impossible differential cryptanalysis of reduced-round Camellia, in: R.M. Avanzi, L. Keliher, F. Sica (Eds.), Selected Areas in Cryptography, in: Lect. Notes Comput. Sci., vol. 5381, Springer, 2008, pp. 442–456.
[40] G. de Meulenaer, F. Gosset, F.-X. Standaert, O. Pereira, On the energy cost of communication and cryptography in wireless sensor networks, in: WiMob, IEEE, 2008, pp. 580–585.
[41] X. Wang, K. Sako (Eds.), Proceedings of the Advances in Cryptology – ASIACRYPT 2012 – 18th International Conference on the Theory and Application of Cryptology and Information Security, Beijing, China, December 2–6, 2012, Lect. Notes Comput. Sci., vol. 7658, Springer, 2012.
[42] D. Pointcheval, T. Johansson (Eds.), Proceedings of the Advances in Cryptology – EUROCRYPT 2012 – 31st Annual International Conference on the Theory and Applications of Cryptographic Techniques, Cambridge, UK, April 15–19, 2012, Lect. Notes Comput. Sci., vol. 7237, Springer, 2012.
[43] A. Mitrokotsa, S. Vaudenay (Eds.), Proceedings of the Progress in Cryptology – AFRICACRYPT 2012 – 5th International Conference on Cryptology in Africa, Ifrance, Morocco, July 10–12, 2012, Lect. Notes Comput. Sci., vol. 7374, Springer, 2012.