ITUbee: A Software Oriented Lightweight Block Cipher

Ferhat Karakoç[1,2], Hüseyin Demirci[1], and A. Emre Harmancı[2]

[1] Tübitak BILGEM UEKAE, 41470, Gebze, Kocaeli, Turkey
{ferhat.karakoc,huseyin.demirci}@tubitak.gov.tr
[2] Istanbul Technical University, Computer Engineering Department, 34469, Maslak, Istanbul, Turkey
harmanci@itu.edu.tr

Abstract. In this paper, we propose a software oriented lightweight block cipher, ITUbee. The cipher is especially suitable for resource constrained devices including an 8-bit microcontroller such as sensor nodes in wireless sensor networks. For a sensor node one of the most important constraints is the low energy consumption because of the limited battery power. Also, the memory on sensor nodes are restricted. We have simulated the performance of ITUbee in the AVR ATtiny45 microcontroller using the integrated development platform Atmel Studio 6. We have evaluated the memory usage and clock cycles needed for an encryption. The number of clock cycles gives a metric for energy consumption. The simulation results show that ITUbee is a competitive block cipher on 8-bit software platforms in terms of energy consumption. Also, less memory requirement of the cipher is remarkable. In addition, we have shown that the attacks which are effective on software oriented lightweight block ciphers can not reduce the 80-bit security level of ITUbee.

Keywords: Lightweight block cipher, cryptanalysis, sensor nodes, AVR ATtiny.

1 Introduction

Lightweight cryptography is needed for the security and privacy demands of the applications where resource constrained devices such as RFID tags and sensor nodes are used. Because of the increase in the usage of these devices in daily life, designing lightweight primitives is getting prominent. Block ciphers are essential primitives in cryptographic applications and therefore there have been several designed lightweight block ciphers. Some of the proposed lightweight block ciphers are DESXL [21], HIGHT [16], KASUMI [1], KATAN [8], KLEIN[14], LBlock [30], LED [15], mCrypton [22], PRESENT [6], PRINCE [7], PRINTcipher [19], SEA [28] and TEA [29]. However, most of the ciphers in this list have a hardware oriented design.

In this paper, we propose a new software oriented lightweight block cipher, ITUbee, for resource constrained devices which include a microcontroller and

G. Avoine and O. Kara (Eds.): LightSec 2013, LNCS 8162, pp. 16–27, 2013.

have a limited battery power such as sensor nodes in wireless sensor networks. To reduce the energy consumption of the cipher we have preferred not to use a key schedule. Also, we have designed the cipher based on a Feistel structure. Generally the ciphers having no key schedule are of SPN structure such as LED and PRINCE. To the best of our knowledge, the only cipher based on Fesitel Structure with a very simple key schedule is GOST [32]. However, a cipher based on Fesitel Structure with no key schedule (or a very simple key schedule) is subject to related key attacks as observed in GOST cipher [20]. To prevent this weakness we have used a new approach in the design of the round function. In every round the round key is injected between two nonlinear operations.

To evaluate the performance of ITUBEE we have simulated the energy consumption and memory usage of the cipher on the Atmel ATtiny45 8-bit microcontoller using Atmel Studio 6. The simulation results show that ITUBEE is an energy efficient block cipher and the memory requirement is very low. We have compared ITUBEE with the ciphers mentioned in a recent work [13] to present its efficiency in terms of energy consumption. Also, we have analyzed the security of the cipher against some attacks applied on software oriented lightweight block ciphers. We claim that ITUBEE offers 80-bit security.

The rest of the paper is organized as follows. In Section 2, we give the definition of ITUBEE. We explain the design rationale in Section 3. In Section 4, we present preliminary security analysis of the cipher. After giving simulation details and performance results of ITUBEE, we compare the performance results with the results of some existing ciphers in Section 5. We conclude the paper with Section 6.

2 Definition of ITUBEE

2.1 Notation

Throughout this paper we use the following notations:

$A\|\|B$	: Concatenation of two bit strings A and B.
K	: 80-bit master key.
K_L	: The left half of the master key.
K_R	: The right half of the master key.
P	: 80-bit plaintext.
P_L	: The left half of the plaintext.
P_R	: The right half of the plaintext.
C	: 80-bit ciphertext.
C_L	: The left half of the ciphertext.
C_R	: The right half of the ciphertext.
RC_i	: The round constant used in the i-th round.

2.2 Definition of the Cipher

The key length and block size of ITUBEE are 80 bits. The cipher has a Feistel structure consisting of 20 rounds and there are key whitening layers at the top and at the bottom of the cipher as illustrated in Figure 1.

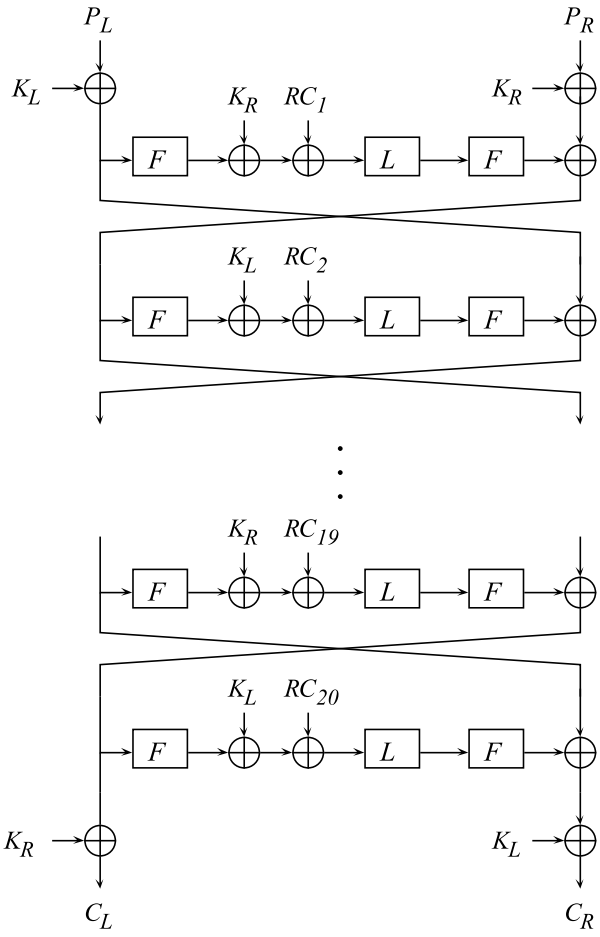

Fig. 1. ITUBEE encryption algorithm

The encryption operation proceeds as follows:

1. $X_1 \leftarrow P_L \oplus K_L$ and $X_0 \leftarrow P_R \oplus K_R$.
2. for $i = 1...20$ do
   (a) if $i \in \{1, 3, ..., 19\}$
      i. $RK \leftarrow K_R$.
   (b) else
      i. $RK \leftarrow K_L$.
   (c) $X_{i+1} \leftarrow X_{i-1} \oplus F(L(RK \oplus RC_i \oplus F(X_i)))$. Note that 16-bit round constant is added to the rightmost 16 bits.
3. $C_L \leftarrow X_{20} \oplus K_R$ and $C_R \leftarrow X_{21} \oplus K_L$.

The definitions of the functions used in the rounds are:

Table 1. Round constants used in 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

- $F(X) = S(L(S(X)))$.
- $S(a\|b\|c\|d\|e) = s[a]\|s[b]\|s[c]\|s[d]\|s[e]$ where a, b, c, d, e are 8-bit values and s is the S-box used in AES [10].
- $L(a\|b\|c\|d\|e) = (e \oplus a \oplus b)\|(a \oplus b \oplus c)\|(b \oplus c \oplus d)\|(c \oplus d \oplus e)\|(d \oplus e \oplus a)$.

The round constants used in ITUBEE are given in Table 1.

The whitening and round keys are derived from the master key directly. $(K_L\|K_R)$ and $(K_R\|K_L)$ are used as whitening keys at the top and at the bottom of the encryption algorithm respectively and for odd rounds K_R is used as round keys while for even rounds K_L is used.

The decryption process is the same as the encryption process except that in the decryption process the key is $(K_R\|K_L)$ and the round constants are used in reversed order.

2.3 Security Goal

ITUBEE offers 80-bit security. Also, the cipher provides the same security level against attacks in related key models as well as in single key attack models. In addition, the cipher has no weak keys.

3 Design Rationale

Power consumption is a major consideration in the design of ITUBEE because of the limited battery power of sensor nodes. Also, we have paid attention on the memory requirement.

We have preferred a Feistel structure which enables us to use the same program code for encryption and decryption processes in a microcontroller which leads to less memory requirement. Also, the elimination of the key schedule reduces the energy consumption and memory requirement. The usage of a Fesitel structure with no key schedule providing security against related key attacks makes ITUBEE different from previously proposed ciphers.

S-boxes satisfy good confusion with less number of operations in microcontrollers. Thus, we have used 8-bit S-boxes to confuse 8 bits with just a table look-up which reduces the power consumption dramatically. In the linear layer we have used a cellular automaton just consisting of 15 XOR operations. To make the proofs for the security of the cipher against differential, linear and

related key differential attacks simple we have constructed a 40-bit substitution box, F function, in a light way which is $S \circ L \circ S$.

The cipher has a Feistel structure and also has no key schedule. These two properties of the cipher makes the cipher weaker for related key differential attacks. One example of such an attack was applied on full GOST [20]. To save the security of the cipher against for such an attack we have injected the round keys between two nonlinear operations (two F functions). To the best of our knowledge this rationale is a not followed before. Also, we have used a linear layer between two F functions to avoid using two consecutive S-boxes.

Some attacks such as reflection [17], slide [4], and slidex [12] use similarities between round functions. To break the similarities between round functions we have used round constants. However, we have reduced the size of the constants to save from the number of operations and memory requirement as in [19]. We have derived that the number of bits of a round constant should be at least 16. If we had used 8-bit round constants then there would be some patterns in the ciphertexts as described in Proposition 1 and 2.

Proposition 1. *F function preserves the pattern for the inputs (a, b, b, a, c) where a, b, c are 8-bit values.*

Proof. The first S-box layer does not change the pattern. After the linear layer the output 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 is the same pattern. Thus, this pattern is not changed by F function.

Proposition 2. *Assume that 1-byte round constants are used in* ITUBEE *and the round constants are added on the rightmost byte of 40-bit value. When the input and the key are of the form $(a \| b \| b \| a \| c \| d \| e \| e \| d \| f)$ then the ciphertext will be in the same form where a, b, c, d, e, f are 1-bytes.*

Proof. P_L, P_R, K_L, K_R have the same pattern as in Proposition 1. Thus, the key additions in the algorithm does not change the pattern. Also, F, L and S-box layers preserve this pattern as presented in Proposition 1. In addition, the round constant addition does not change the pattern because the constant changes only the rightmost byte which does not affect the pattern. As a result, C_L and C_R will have the same pattern and this is independent of the number of rounds.

Also, we have chosen i-th round constant as $(0x15 - i) \| (0x29 - i)$ not to use any memory for the constants in the case where only encryption process is required in the microcontroller.

We have observed that the maximum number of rounds that cryptanalytic attacks can be applied is around 10. To have a high security margin we have decided the number of rounds as 20.

4 Security Analysis

4.1 Differential and Linear Cryptanalysis

The differential and linear cryptanalysis are the mostly used cryptanalysis techniques [3,25]. In these techniques, nonlinear operations in an encryption process

are treated as linear operations with a probability to model the whole cipher as a linear algorithm.

The only nonlinear part in ITUBEE is the S-box operation. Thus, counting the number of active S-boxes in differential and linear characteristics is the main work of the differential and linear cryptanalysis of the cipher. The branch number of the linear layer in F is 4, that means at least 4 S-boxes are active when one F is active. For one round, if the left half of the input is active then there will be 2 active F functions. In the Feistel Structure the left halves of the inputs of at least 2 rounds out of 3 consecutive rounds have differences if the function which produces the output to add to the right half is one-to-one. In our cipher, this function is one-to-one, so we can say that we have at least 4 active F functions that is 16 active S-boxes for 3 consecutive rounds. The S-box used in ITUBEE is the AES S-box and the best probability for one input-output difference is 2^{-6} and the best linear bias for an input-output mask is 2^{-4}. Thus, for consecutive 3 rounds the best probability of a differential characteristic and the best linear bias of a linear characteristic will be $(2^{-6})^{16} = 2^{-96}$ and $2^{15} \times (2^{-4})^{16} = 2^{-49}$ respectively which are not usable in differential and linear attacks. Therefore, it seems that differential and linear attacks can not be applied on 20-round ITUBEE.

However, to give a proof against the differential and linear attacks it is not sufficient to count the number of active S-boxes in the characteristics because of the differential and the linear hull effects.

To analyze the differential effects on our cipher we have focused on the differential probabilities for the F function. We have seen that while one active F function has at least 4 active S-boxes which leads to the probability $(2^{-6})^4 = 2^{-24}$, the F function can have differentials having a greater probability than 2^{-24}. The reason is the following summation of the probabilities of the characteristics:

$$Pr(\Delta X \xrightarrow{F} \Delta Y) = \sum_{\Delta Z} Pr(\Delta X \xrightarrow{S} \Delta Z) \times Pr(L(\Delta Z) \xrightarrow{S} \Delta Y).$$

This effect can increase the probability of a differential for the F function but the probability will be less than 2^{-17}. In the case where only 4 bytes are active totally in the input and output of the F function, there will only be one free active byte which can take any difference. This free active byte can take at most 2^7 different values. Thus, the summation of the probabilities of differential characteristics which leads to a differential for the F function will be less than $(2^{-6})^4 \times 2^7 = 2^{-17}$.

In consecutive 6 rounds there will be at least 8 active F functions and therefore the probability of a differential characteristic for 6 rounds will be less than $(2^{-17})^8 = 2^{-136}$. However, it is necessary to consider the differential effect on consecutive two F functions. To see this effect we have performed experiments on a toy version of F functions. In this version the S-box is replaced with a 3-bit S-box and the word size of 8-bit is replaced with 3-bit and the other operations are same. In this version the maximum probability of F function is less than or equal to $(2^{-2})^4 \times 2^2 = 2^{-6}$. The experiment result shows that the maximum probability of consecutive two F functions is not bigger than 2^{-6}. As a result

of this experiment, we assume that maximum differential probability for consecutive two F functions is not bigger than 2^{-17}. For 8 rounds there is at least 5 active rounds and so this leads to a probability smaller than $(2^{-17})^5 = 2^{-85}$ which is not usable in a classical differential attack. Also, note that the differentials which has a maximum probability for two consecutive F functions are key dependent so this leads to another difficulty for these type of attacks.

4.2 Meet-in-the-Middle Type Attacks

Each key bit affects all bits of the output after consecutive 3 rounds. Thus, we assume that the basic meet-in-the-middle attacks are not applicable to our cipher. Recently, there have been some extensions of the basic meet-in-the-middle attacks such as the multidimensional meet-in-the-middle attack [33], the biclique attack [5]. In the case of independent biclique type attacks, let us assume that the key whitenings does not exist. The maximum number of rounds on which a biclique can be constructed is 2. Also, in the meet-in-the-middle step of this attack the number of F functions in the recalculation step which is done for the whole key space is about 32. Thus, the complexity of such an attack is approximately $\frac{32\times 2^{80}}{40} \approx 2^{-79.678}$ so this attack can not reduce the security level of ITUBEE more than 1-bit. The multidimensional meet-in-the-middle attack is usable if the key length of the cipher is bigger than the block size. For ITUBEE the block size and key length are the same. We conclude that also this attack does not threat the security level of our cipher.

4.3 Related Key Differential Attacks

In the related key attack model, the adversary is able to collect plaintext and ciphertext pairs under related keys. In our algorithm, we divide the master key into two parts K_L and K_R and we use these parts between F functions in the rounds. It is trivial to see that when there is a difference in the key used between F functions, then at least one of the F functions will be active. In the best case for the adversary, the difference will be only one part of the key K_L or K_R. As a result, in two consecutive rounds there exists at least one active F function in the case of the related key attack. The probabilities of differentials for the F function is less than 2^{-17} as given in the Section 4.1. Thus, for 10 consecutive rounds the probabilities will be less than $(2^{-17})^5 = 2^{-85}$ which is not usable in an attack.

4.4 Impossible Differential Attacks

One of the most powerful attacks on lightweight block ciphers is the impossible differential attack [2] which have been applied on many lightweight ciphers [9,18,23,24,27,31]. We could not find any impossible differential characteristic for 6 or more rounds. We conclude that this attack technique is not applicable to our cipher.

4.5 Self-similarity Attacks

The sel-similarty attacks such as reflection [17], slide [4], and slidex [12] use similarities of round functions. For ITUBEE the round functions are very similar due to the non-existence of the key schedule. The only part in the cipher which prevent the cipher from these attacks is the round constant addition. We believe that because of the round constant these attacks can not be applied to ITUBEE.

5 Simulation Results

We have written the code of ITUBEE in assembly and simulated the energy consumption and memory usage of it on the Atmel 8-bit AVR ATtiny45 RISC-based microcontroller using Atmel Studio 6. The microcontroller has a Harvard architecture in which the instruction and data memory are separated. The instruction and data memory are a 4-kB Flash memory and 256-byte static RAM, respectively. In the implementation of our cipher we have stored the 8-bit S-box used in the cipher in the instruction memory. Also, we have used CPU registers for all internal variables and we have not used any SRAM except for the plaintext/ciphertext and master key. To compare the performance of our cipher with some other lightweight ciphers we present the memory requirement of the encryption process and the number of clock cycles needed for an encryption operation of some ciphers in Table 2. Note that the implementations in [13] were also on an ATtiny45 microcontroller. Also, we give the number of clock cycles per one byte in the table dividing the number of cycles for an encryption to the block size in bytes. In addition, we present the product of the number of clock cycles per one byte with the memory requirement of the ciphers in the table to give another metric to compare the performance results of the ciphers.

The table demonstrates that an ITUBEE encryption process is performed in less clock cycles than the other ciphers. The number of clock cycles is strongly correlated with the energy consumption [11,13]. As a result, it can be extracted from the table that ITUBEE is the encryption algorithm which has the least energy consumption in the list. 716 byte memory requirement of the cipher is also remarkable. In the case of using less memory, the energy consumption will increase but it is still remarkable.

6 Conclusion

We have proposed a software oriented lightweight block cipher named ITUBEE for applications such as wireless sensor networks consisting of low-power nodes including an 8-bit microcontroller. We have used a Feistel structure with no key schedule to reduce the energy consumption. To prevent the cipher from related key attacks we have used the round key addition between two nonlinear layers in each round. We have simulated the performance of the cipher in terms of energy consumption and memory usage on the 8-bit ATtiny45 microcontroller. We have shown that ITUBEE consumes less energy than the ciphers whose performance

Table 2. Performance results of some lightweight ciphers

Cipher	Block size [bits]	Key size [bits]	Memory [bytes]	Clock cycles per one enc.	Clock cycles per one byte	Cycle × Memory
AES [13]	128	128	1689	4557	284	479676
DESXL [13]	64	184	868	84602	10575	9179100
HIGHT [13]	64	128	434	19503	2437	1057658
IDEA [13]	64	128	1068	≈ 8250	1031	1101108
KASUMI [13]	64	128	1288	11939	1492	1921696
KATAN [13]	64	80	356	72063	9007	3206492
KLEIN [13]	64	80	1286	6095	761	978646
mCrypton [13]	64	96	1104	16457	2057	2270928
NOEKEON [13]	128	128	396	23517	1469	581724
PRESENT [13]	64	80	1018	11342	1417	1442506
SEA [13]	96	96	450	41604	3467	1560150
TEA [13]	64	128	672	7408	926	622272
LBlock [30]	64	80	not given	3955	494	-
ITUBEE [this paper] cycle optimized	80	80	716	2607	261	186876
ITUBEE [this paper] memory optimized	80	80	586	2937	294	172284

results are given in a recent work [13] while the memory usage of ITUBEE is also remarkable. In addition, we have analyzed the security of the cipher against some attacks which are effective on software oriented lightweight block ciphers and we have concluded that these attacks can not reduce the 80-bit security level of ITUBEE.

Acknowledgments. We thank to Özkan Boztaş and Cevat Manap for their helpful comments on the design of the cipher. We also thank to anonymous reviewers for their valuable comments which helped us to improve the quality of this paper.

References

1. ETSI. TS 135 202 V7.0.0: Universal Mobile Telecommunications System (UMTS); Specification of the 3GPP confidentiality and integrity algorithms; Document 2: KASUMI specification (3GPP TS 35.202 version 7.0.0 Release 7), `http://www.etsi.org`
2. Biham, E., Biryukov, A., Shamir, A.: Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials. In: Stern, J. (ed.) EUROCRYPT 1999. LNCS, vol. 1592, pp. 12–23. Springer, Heidelberg (1999)
3. Biham, E., Shamir, A.: Differential Cryptanalysis of DES-like Cryptosystems. In: Menezes, A., Vanstone, S.A. (eds.) CRYPTO 1990. LNCS, vol. 537, pp. 2–21. Springer, Heidelberg (1991)
4. Biryukov, A., Wagner, D.: Slide attacks. In: Knudsen, L.R. (ed.) FSE 1999. LNCS, vol. 1636, pp. 245–259. Springer, Heidelberg (1999)

5. Bogdanov, A., Khovratovich, D., Rechberger, C.: Biclique Cryptanalysis of the Full AES. In: Lee, D.H., Wang, X. (eds.) ASIACRYPT 2011. LNCS, vol. 7073, pp. 344–371. Springer, Heidelberg (2011)
6. Bogdanov, A., Knudsen, L.R., Leander, G., Paar, C., Poschmann, A., Robshaw, M.J.B., Seurin, Y., Vikkelsoe, C.: PRESENT: An Ultra-Lightweight Block Cipher. In: Paillier, P., Verbauwhede, I. (eds.) CHES 2007. LNCS, vol. 4727, pp. 450–466. Springer, Heidelberg (2007)
7. Borghoff, J., Canteaut, A., Güneysu, T., Kavun, E.B., Knezevic, M., Knudsen, L.R., Leander, G., Nikov, V., Paar, C., Rechberger, C., Rombouts, P., Thomsen, S.S., Yalçın, T.: PRINCE - A Low-Latency Block Cipher for Pervasive Computing Applications - Extended Abstract. In: Wang, X., Sako, K. (eds.) ASIACRYPT 2012. LNCS, vol. 7658, pp. 208–225. Springer, Heidelberg (2012)
8. De Cannière, C., Dunkelman, O., Knežević, M.: KATAN and KTANTAN — A Family of Small and Efficient Hardware-Oriented Block Ciphers. In: Clavier, C., Gaj, K. (eds.) CHES 2009. LNCS, vol. 5747, pp. 272–288. Springer, Heidelberg (2009)
9. Chen, J., Wang, M., Preneel, B.: Impossible Differential Cryptanalysis of the Lightweight Block Ciphers TEA, XTEA and HIGHT. In: Mitrokotsa, Vaudenay (eds.) [26], pp. 117–137
10. Daemen, J., Rijmen, V.: The Design of Rijndael: AES - The Advanced Encryption Standard. Springer (2002)
11. de Meulenaer, G., Gosset, F., Standaert, F.-X., Pereira, O.: On the Energy Cost of Communication and Cryptography in Wireless Sensor Networks. In: WiMob, pp. 580–585. IEEE (2008)
12. Dunkelman, O., Keller, N., Shamir, A.: Minimalism in Cryptography: The Even-Mansour Scheme Revisited. In: Pointcheval, D., Johansson, T. (eds.) EUROCRYPT 2012. LNCS, vol. 7237, pp. 336–354. Springer, Heidelberg (2012)
13. Eisenbarth, T., Gong, Z., Güneysu, T., Heyse, S., Indesteege, S., Kerckhof, S., Koeune, F., Nad, T., Plos, T., Regazzoni, F., Standaert, F.-X., van Oldeneel tot Oldenzeel, L.: Compact Implementation and Performance Evaluation of Block Ciphers in ATtiny Devices. In: Mitrokotsa, Vaudenay (eds.) [26], pp. 172–187
14. Gong, Z., Nikova, S., Law, Y.W.: KLEIN: A New Family of Lightweight Block Ciphers. In: Juels, A., Paar, C. (eds.) RFIDSec 2011. LNCS, vol. 7055, pp. 1–18. Springer, Heidelberg (2012)
15. Guo, J., Peyrin, T., Poschmann, A., Robshaw, M.: The LED Block Cipher. In: Preneel, B., Takagi, T. (eds.) CHES 2011. LNCS, vol. 6917, pp. 326–341. Springer, Heidelberg (2011)
16. Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B., Lee, C., Chang, D., Lee, J., Jeong, K., Kim, H., Kim, J., Chee, S.: HIGHT: A New Block Cipher Suitable for Low-Resource Device. In: Goubin, L., Matsui, M. (eds.) CHES 2006. LNCS, vol. 4249, pp. 46–59. Springer, Heidelberg (2006)
17. Kara, O.: Reflection Cryptanalysis of Some Ciphers. In: Chowdhury, D.R., Rijmen, V., Das, A. (eds.) INDOCRYPT 2008. LNCS, vol. 5365, pp. 294–307. Springer, Heidelberg (2008)
18. Karakoç, F., Demirci, H., Emre Harmanci, A.: Impossible Differential Cryptanalysis of Reduced-Round LBlock. In: Askoxylakis, I., Pöhls, H.C., Posegga, J. (eds.) WISTP 2012. LNCS, vol. 7322, pp. 179–188. Springer, Heidelberg (2012)
19. Knudsen, L., Leander, G.R., Poschmann, A., Robshaw, M.J.B.: PRINTcipher: A Block Cipher for IC-Printing. In: Mangard, S., Standaert, F.-X. (eds.) CHES 2010. LNCS, vol. 6225, pp. 16–32. Springer, Heidelberg (2010)

20. Ko, Y., Hong, S., Lee, W., Lee, S., Kang, J.-S.: Related Key Differential Attacks on 27 Rounds of XTEA and Full-Round GOST. In: Roy, B.K., Meier, W. (eds.) FSE 2004. LNCS, vol. 3017, pp. 299–316. Springer, Heidelberg (2004)
21. Leander, G., Paar, C., Poschmann, A., Schramm, K.: New Lightweight DES Variants. In: Biryukov, A. (ed.) FSE 2007. LNCS, vol. 4593, pp. 196–210. Springer, Heidelberg (2007)
22. Lim, C.H., Korkishko, T.: mCrypton – A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors. In: Song, J.-S., Kwon, T., Yung, M. (eds.) WISA 2005. LNCS, vol. 3786, pp. 243–258. Springer, Heidelberg (2006)
23. Liu, Y., Gu, D., Liu, Z., Li, W.: Impossible Differential Attacks on Reduced-Round LBlock. In: Ryan, M.D., Smyth, B., Wang, G. (eds.) ISPEC 2012. LNCS, vol. 7232, pp. 97–108. Springer, Heidelberg (2012)
24. Liu, Y., Gu, D., Liu, Z., Li, W.: Improved Results on Impossible Differential Cryptanalysis of Reduced-Round Camellia-192/256. Journal of Systems and Software 85(11), 2451–2458 (2012)
25. Matsui, M.: Linear Cryptanalysis Method for DES Cipher. In: Helleseth, T. (ed.) EUROCRYPT 1993. LNCS, vol. 765, pp. 386–397. Springer, Heidelberg (1994)
26. Mitrokotsa, A., Vaudenay, S. (eds.): AFRICACRYPT 2012. LNCS, vol. 7374. Springer, Heidelberg (2012)
27. Özen, O., Varıcı, K., Tezcan, C., Kocair, Ç.: Lightweight Block Ciphers Revisited: Cryptanalysis of Reduced Round PRESENT and HIGHT. In: Boyd, C., González Nieto, J. (eds.) ACISP 2009. LNCS, vol. 5594, pp. 90–107. Springer, Heidelberg (2009)
28. Standaert, F.-X., Piret, G., Gershenfeld, N., Quisquater, J.-J.: SEA: A Scalable Encryption Algorithm for Small Embedded Applications. In: Domingo-Ferrer, J., Posegga, J., Schreckling, D. (eds.) CARDIS 2006. LNCS, vol. 3928, pp. 222–236. Springer, Heidelberg (2006)
29. Wheeler, D.J., Needham, R.M.: TEA, a Tiny Encryption Algorithm. In: Preneel, B. (ed.) FSE 1994. LNCS, vol. 1008, pp. 363–366. Springer, Heidelberg (1995)
30. Wu, W., Zhang, L.: LBlock: A Lightweight Block Cipher. In: Lopez, J., Tsudik, G. (eds.) ACNS 2011. LNCS, vol. 6715, pp. 327–344. Springer, Heidelberg (2011)
31. Wu, W., Zhang, L., Zhang, W.: Improved Impossible Differential Cryptanalysis of Reduced-Round Camellia. In: Avanzi, R.M., Keliher, L., Sica, F. (eds.) SAC 2008. LNCS, vol. 5381, pp. 442–456. Springer, Heidelberg (2009)
32. Zabotin, I.A., Glazkov, G.P., Isaeva, V.B.: Cryptographic Protection for Information Processing Systems. Cryptographic Transformation Algorithm. Government Standard of the USSR, GOST 28147-89 (1989)
33. Zhu, B., Gong, G.: Multidimensional meet-in-the-middle attack and its applications to katan32/48/64. Cryptology ePrint Archive, Report 2011/619 (2011), http://eprint.iacr.org/

A AES S-Box

```
s[256] = {
0x63, 0x7C, 0x77, 0x7B, 0xF2, 0x6B, 0x6F, 0xC5, 0x30, 0x01, 0x67, 0x2B, 0xFE, 0xD7, 0xAB, 0x76
0xCA, 0x82, 0xC9, 0x7D, 0xFA, 0x59, 0x47, 0xF0, 0xAD, 0xD4, 0xA2, 0xAF, 0x9C, 0xA4, 0x72, 0xC0
0xB7, 0xFD, 0x93, 0x26, 0x36, 0x3F, 0xF7, 0xCC, 0x34, 0xA5, 0xE5, 0xF1, 0x71, 0xD8, 0x31, 0x15
0x04, 0xC7, 0x23, 0xC3, 0x18, 0x96, 0x05, 0x9A, 0x07, 0x12, 0x80, 0xE2, 0xEB, 0x27, 0xB2, 0x75
0x09, 0x83, 0x2C, 0x1A, 0x1B, 0x6E, 0x5A, 0xA0, 0x52, 0x3B, 0xD6, 0xB3, 0x29, 0xE3, 0x2F, 0x84
0x53, 0xD1, 0x00, 0xED, 0x20, 0xFC, 0xB1, 0x5B, 0x6A, 0xCB, 0xBE, 0x39, 0x4A, 0x4C, 0x58, 0xCF
0xD0, 0xEF, 0xAA, 0xFB, 0x43, 0x4D, 0x33, 0x85, 0x45, 0xF9, 0x02, 0x7F, 0x50, 0x3C, 0x9F, 0xA8
```

```
0x51, 0xA3, 0x40, 0x8F, 0x92, 0x9D, 0x38, 0xF5, 0xBC, 0xB6, 0xDA, 0x21, 0x10, 0xFF, 0xF3, 0xD2
0xCD, 0x0C, 0x13, 0xEC, 0x5F, 0x97, 0x44, 0x17, 0xC4, 0xA7, 0x7E, 0x3D, 0x64, 0x5D, 0x19, 0x73
0x60, 0x81, 0x4F, 0xDC, 0x22, 0x2A, 0x90, 0x88, 0x46, 0xEE, 0xB8, 0x14, 0xDE, 0x5E, 0x0B, 0xDB
0xE0, 0x32, 0x3A, 0x0A, 0x49, 0x06, 0x24, 0x5C, 0xC2, 0xD3, 0xAC, 0x62, 0x91, 0x95, 0xE4, 0x79
0xE7, 0xC8, 0x37, 0x6D, 0x8D, 0xD5, 0x4E, 0xA9, 0x6C, 0x56, 0xF4, 0xEA, 0x65, 0x7A, 0xAE, 0x08
0xBA, 0x78, 0x25, 0x2E, 0x1C, 0xA6, 0xB4, 0xC6, 0xE8, 0xDD, 0x74, 0x1F, 0x4B, 0xBD, 0x8B, 0x8A
0x70, 0x3E, 0xB5, 0x66, 0x48, 0x03, 0xF6, 0x0E, 0x61, 0x35, 0x57, 0xB9, 0x86, 0xC1, 0x1D, 0x9E
0xE1, 0xF8, 0x98, 0x11, 0x69, 0xD9, 0x8E, 0x94, 0x9B, 0x1E, 0x87, 0xE9, 0xCE, 0x55, 0x28, 0xDF
0x8C, 0xA1, 0x89, 0x0D, 0xBF, 0xE6, 0x42, 0x68, 0x41, 0x99, 0x2D, 0x0F, 0xB0, 0x54, 0xBB, 0x16
}
```

B Test Vectors

Plaintext	Key	Ciphertext
00000000000000000000	00000000000000000000	471330577984cbecf6c8
01000000000000000000	00000000000000000080	761b8299b3f6a99f0838
6925278951fbf3b25ccc	c538bd9289822be43363	c42e0f48cd5a87d0055f