# Survey and Benchmark of Lightweight Block Ciphers for Wireless Sensor Networks*

Mickaël Cazorla, Kevin Marquet and Marine Minier

Université de Lyon, INRIA
INSA-Lyon, CITI-INRIA, F-69621, Villeurbanne, France
firstname.name@insa-lyon.fr

**Abstract.** For security applications in wireless sensor networks (WSNs), choosing best algorithms in terms of energy-efficiency and of small memory requirements is a real challenge because the sensor networks must be autonomous. In [17, 35], the authors have benchmarked on a dedicated platform some block-ciphers and have deduced the best candidates to use in the context of small embedded platforms. This article proposes to study on a dedicated platform of sensors most of the recent lightweight block ciphers as well as some conventional block ciphers. First, we describe the design of the chosen block ciphers with a security summary and we then present some implementation tests performed on our platform.

**Keywords:** lightweight block ciphers, sensors, benchmarks.

## Introduction

Wireless Sensor Networks (WSNs) are composed of numerous low-cost, low-energy sensor nodes communicating at short distance through wireless links. Sensors are densely deployed to collect and transmit data of the physical world to one or few destinations called the sinks using multi-hop routing protocols. Wireless sensor networks can be really useful in many civil and military areas for collecting, processing and monitoring environmental data. A sensor node contains an integrated sensor, a microprocessor, some memories, a transmitter and an energy battery. Despite the relative simplicity of its basic components, sensor networking offers a great diversity: various hardwares (MicaZ, Telos, SkyMote, AVR or TI micro-controllers), various radio and physical layers (868MHz and 2,4GHz) using different types of modulations, various OS (TinyOS, Contiki, FreeRTOS, JITS), various constraints (real-time, energy, memory or processing), various applications (military or civil uses).

In such a context, a specific care must be invested in the design of the applications, communication protocols, operating systems and of course security protocols that will be used. Lots of protocols have been proposed to enforce the security offered by sensor networks. Despite the increasing request in this new area of research, few articles present results of real software implementations or benchmarks concerning the security primitives which can be used in sensor networks. In [17, 35], the authors present such results. In [35], the authors present benchmark results on a MSP430, a TI 16 bits microcontroller, comparing the most famous block ciphers (including AES, MISTY1, Skipjack,...) and the different possible modes of operations. In [17], the authors present benchmark results on a ATtiny device, a 8 bits microcontroller, of 12 block ciphers, 8 lightweight block ciphers and 4 conventional block ciphers. They also introduce a comparison metric that takes into account both the code size and the cycle count.

This article proposes to theoretically sum-up the security provided by most of the recent lightweight block ciphers and some conventional block ciphers and presents some implementation tests performed on a MSP430[1], a TI 16 bits microcontroller which is the corner stone of the nodes WSN430[2] used in the deployed Senslab platform[3] [16].

This paper is organized as follows: Section 1 presents several block ciphers, evaluates their current security based on the most recent results and precises when required the implementation

* This work was partially supported by the French National Agency of Research: ANR-11-INS-011.

[1] http://www.ti.com/product/msp430f1611

[2] http://perso.ens-lyon.fr/eric.fleury/Upload/wsn430-docbook/

[3] http://www.senslab.info/

choices. Section 2 presents the dedicated platform and describes the methodology used to perform our benchmarks. Section 3 provides our results and our analysis concerning the benchmarking whereas Section 4 concludes this paper.

## 1 The studied block ciphers

Our benchmarks concern 17 block ciphers, 12 are lightweight and 5 are conventional block ciphers. Studied block ciphers are listed in Table 1 in alphabetic order.

The main differences between the conventional block ciphers and the lightweight block ciphers are centered on: the block size which is in general 32, 48 or 64 bits for a lightweight block cipher and equal to 64 or 128 bits for a conventional block cipher; the same remark also holds for the different possible key sizes (smaller for lightweight block ciphers); Lightweight block ciphers also rely more on elementary operations (such as binary XOR, binary AND, etc.) leading in an increase of required number of rounds; Lightweight block ciphers generally extremely simplify the key schedule due to memory requirements.

In this section, we give a quick overview of each implemented block cipher from a design point of view (without describing the key schedule) and from a cryptanalytic point of view (we limit our state of art in the case of unknown key settings and of related key settings, we do not describe attacks in the known or chosen key settings). We also provide some words about the way we implemented them when required.

| NAME ($Nb/Nk$) | Reference | Struct. | Nb rounds |
|---|---|---|---|
| AES-128* (128/128) | [18] | SPN | 10 |
| CLEFIA-128* (128/128) | [48] | Feistel | 18 |
| DESXL (64/184) | [37] | Feistel | 16 |
| HIGHT (64/128) | [23] | Feistel | 32 |
| IDEA* (64/128) | [34] | Lai-Massey | 8.5 |
| KATAN & KTANTAN (32, 48, 64/80) | [9] | Stream | 254 |
| KLEIN (64/64, 80 and 96) | [19] | SPN | 12, 16, 20 |
| LBLOCK (64/80) | [57] | Feistel | 32 |
| LED (64/64 and 128) | [20] | SPN | 32/48 |
| mCrypton (64/64, 96 and 128) | [39] | SPN | 12 |
| MIBS (64/64 and 80) | [25] | Feistel | 32 |
| Noekeon* (128/128) | [14] | SPN | 16 |
| Piccolo (64/80 and 128) | [47] | Feistel | 25/31 |
| PRESENT (64/80 and 128) | [8] | SPN | 31 |
| TEA & XTEA (64/128) | [56] | Feistel | 64 |
| TWINE (64/ 80 and 128) | [51] | Feistel | 36 |
| SEA (96/96,...) | [50] | Feistel | Var. |
| SKIPJACK* (64/80) | [44] | Feistel | 32 |

**Table 1.** Studied block ciphers. $Nb$ means the size of the input/output block in bits; $Nk$ means the size of the key in bits. * designates the conventional block ciphers in opposite to lightweight block ciphers.

***AES-128*** The AES is the current block cipher standard [18] designed by J. Daemen and V. Rijmen in 1997 and chosen as a standard in 2000. It is the most used block cipher. The AES is an iterated block cipher based on a SPN structure that ciphers block of size 128 bits under 128, 192 or 256 bits keys. We focus here on the case of AES-128 that cipher 128 bits blocks under a key of length 128 bits. This AES version is composed of 10 rounds that repeat four elementary mappings (SubBytes, ShiftRows, MixColumns and AddRoundKey) on blocks seen as 4 × 4 byte matrices.
*Security* The main security result against the AES-128 is a biclique cryptanalysis due to A. Bogdanov, D. Khovratovich, C. Rechberger [7]. It improves the key exhaustive search using particular relations linking together the keys through the key schedule and some bytes on internal states. The time complexity of this attack on the full AES-128 version is equal to $2^{126.2}$ AES-128 encryptions whereas

the memory requirements are small and the amount of data is equal to $2^{88}$. The other interesting cryptanalytic result is due to P. Derbez, P.-A. Fouque and J. Jean in [15] that provide a dedicated meet-in-the-middle attack on 7 rounds of AES-128 where data/time/memory complexities are below $2^{100}$.

*Implementation* Our implementation is without tables for the ShiftRows and MixColumns operations and uses a matrix of bytes.

***CLEFIA-128*** CLEFIA is a conventional block cipher designed by Sony and described in [48]. It has been created to achieve good results both in hardware and software. It ciphers block of length 128 bits under keys of length 128, 192 or 256 bits. This cipher is based on a generalized Feistel structure with 4 data lines with two 32-bit F-functions per round. The number of rounds depends on the key length and is equal to 18, 22 or 26 according the key size. The two F-functions call 2 different 8 bits Sboxes followed by a diffusion matrix multiplication inspired from the AES MixColumns operation.

*Security* In [53], the authors present impossible differential cryptanalysis against CLEFIA. With this method, they could build impossible differential attack against CLEFIA reduced to 12 rounds for a 128 bits key with a time complexity equal to $2^{119}$ encryptions. For key lengths of 192 bits and 256 bits, they manage to apply impossible differential attacks to 13-round and 14-round CLEFIA with time complexities of $2^{146}$ and $2^{212}$ encryptions. In [52], the author proposes a new kind of cryptanalysis called improbable differential cryptanalysis. By using this expansion method, the author cryptanalyzed 13, 14, and 15-round CLEFIA for the key sizes of length 128, 192, and 256 bits, respectively with a time complexity slightly lower than the exhaustive key search.

*Implementation* To store the cipher and the key, we use arrays of 8-bit numbers. The rest of the implementation follows the original specification.

***DESL and DESXL*** DESL and DESXL are two lightweight variants of the Data Encryption Standard proposed by G. Leander, C. Paar, A. Poschmann and K. Schramm in [37]. The main idea is to simplify the DES round function using a single S-box instead of 8 and to discard the initial and final permutation of the DES to limit the size of the hardware implementation. So DESL iterates 16 rounds of a classical Feistel network and takes a block of size 64 bits as the DES under a key of size 56 bits whereas DESXL uses, as DESX, a whitening method to reinforce the security with a key of length 184 bits with 64 bits blocks. We only consider DESXL in our implementations for clear security reasons.

*Security* Up to our knowledge, no attack has been exhibited against DESL and DESXL. It seems logical as the authors of [37] repaired all the known weaknesses of the DES, especially by chosen a new well suited Sbox.

*Implementation* To store the block to cipher and the key, we use arrays of 8-bit numbers. We used particular tables to simplify the key schedule.

***HIGHT*** HIGHT is a dedicated lightweight block cipher proposed at CHES 2006 [23]. It takes blocks of size 64 bits under keys of length 128 bits iterating on 32 rounds a modified 8-branch Feistel network where the XOR operation is sometimes replaced by a modular addition. The two internal functions of the Feistel network consist in XOR operations combined with left or right rotations.

*Security* One of the best known attack in the unknown key settings against HIGHT is an impossible differential attack proposed in [11] against 27 rounds of HIGHT with a complexity slightly lower than the exhaustive search. In [22], the authors propose a biclique attack against the full rounds of HIGHT with a computational complexity of $2^{126.4}$ encryptions, faster than exhaustive search based on 8-round bicliques. In [33], the authors propose a related key attack against the full rounds of HIGHT faster than an exhaustive key search using 4 related keys.

*Implementation* The subkeys generation uses constants produced by a LFSR. We used a 128-bytes table to store those constants. The key and the the block to cipher are stored in tables of 8-bit numbers.

***IDEA*** IDEA [34] is a conventional block cipher that uses 64 bits blocks with a 128 bits key. It is composed of 8.5 identical rounds. It is based on the Lai-Massey scheme and interleaves operations on 16 bits words from different groups (modular additions, modular multiplications and XORs). It is one of the most widely used block ciphers, due to its inclusion in several cryptographic packages, such as PGP.
*Security* Since its publication, IDEA resisted intensive cryptanalytic efforts and no attack on the full IDEA version exists. The most significant attacks are a 6 rounds attack faster than exhaustive key search that exploits the weak key-schedule algorithm of IDEA, and combines Square-like techniques with linear cryptanalysis [5]. In [28], the authors apply and extend the biclique framework to IDEA and for the first time describe an approach to noticeably speed-up key-recovery for the full 8.5 rounds IDEA. In addition to these attacks, three relatively large and easily detectable classes of weak keys were found [21, 6].

*Implementation* Due to the complexity of the extended Euclidean algorithm, the subkeys are generated and put in a 52-entry table. To store the block to cipher and the key, we use arrays of 8-bit numbers.

***KATAN and KTANTAN*** KATAN and KTANTAN are two block ciphers based on stream ciphers design proposed at CHES 2009 [9]. They both take as input blocks of sizes 32, 48 or 64 bits under a 80 bits key and iterate during 254 rounds a kind of stream ciphers composed of two LFSRs and non-linear operations. The KATAN and KTANTAN differ from their key schedules: in KATAN, the 80 bits key is loaded into a register which is repeatedly clocked, whereas in KTANTAN, the key is burnt (i.e., fixed) and the only possible "flexibility" is the choice of subkey bits.
*Security* Against KTANTAN, the authors of [55] propose a meet-in-the-middle attack that recovers the 80-bit secret key of the full rounds KTANTAN-{32, 48, 64} at time complexity of $2^{72.9}$, $2^{73.8}$ and $2^{74.4}$ respectively, each requiring 4 chosen-plaintexts. The best attack against KATAN is a conditional differential cryptanalysis described in [30, 31]. In [30], the authors propose a conditional differential cryptanalysis with a practical complexity in the single key settings against KATAN-{32, 48,64} on respectively 78, 70 and 68 rounds. In [31], the same authors extend their previous results in the related key settings against KATAN-{32, 48,64} on respectively 120, 103 and 90 rounds always with a practical complexity in all cases.
*Implementation* These variants makes use of two round functions. At each round, the choice of using one function or the other is made using a precomputed bit *IR*. The difference between those functions is just an additional xor in the case where this bit is '1'. Therefore, we use a constant table in which each i-th cell contains a value that must be used in the i-th round. This value is a bitfield full of '1' when the xor must be applied, and '0' when not (e.g. in the case of a 32-bits security key, a 64-bit int is set to 0xFFFFFFFFFFFFFFFF). We store the block to cipher and the key in tables of 64-bit numbers.

***KLEIN*** In [19], the authors propose a new lightweight block cipher called KLEIN. It ciphers blocks of size 64 bits under a key of length 64, 80 and 96 bits with a variable number of rounds equal to 12, 16 or 20. It is based on a SPN structure which mixes together elementary operations coming from the AES and from PRESENT.
*Security* In [2], the authors exploit the existence of differentials of unexpectedly high probability coming from the combination of the 4 bits Sbox and the byte-oriented MixColumns operation to construct practical and experimentally verified chosen-plaintext key-recovery attacks on up to 8 rounds of KLEIN-64.
*Implementation* To store the block to cipher and the key, we use arrays of 8-bit numbers.

***LBLOCK*** LBLOCK has been proposed in [57]. It ciphers blocks of size 64 bits under keys of size 80 bits using 32 rounds of a modified Feistel network. The round function is composed of a subkey addition, 8 Sboxes applied in parallel followed by a 4 bits permutation.
*Security* Even if LBLOCK is a very recent algorithm, 5 papers present cryptanalytic results against it. Three of them proposed improved results on impossible differential attacks on 23 rounds of

LBLOCK. [54] proposes a biclique attack against a full round version of LBLOCK with a complexity slightly lower than exhaustive key search and proposes a modified key schedule algorithm to prevent this attack from happening. In [49], the authors propose to apply a new cryptanalytic method called zero-correlation linear attack against LBLOCK and they managed to mount a 22 rounds attack with a complexity slightly lower than the exhaustive key search that works for the both key schedule versions of LBLOCK.
*Implementation* To store the block to cipher and the key, we use arrays of 8-bit numbers.

***LED*** LED is a lightweight block cipher [20] that ciphers 64 bits blocks under keys of length 64 or 128 bits (and could be adapted for a 80 bits key). The number of rounds is 32 for a 64 bits key and 48 for a 128 bits key. Each block to cipher is represented by a 4 × 4 nibble matrix. LED is a SPN block cipher that uses the same inner transformations than the AES but on nibbles and optimized for hardware applications. One of the main originality of LED is the absence of key schedule, instead the key is xored every 4 rounds. This absence is compensated by an increased number of rounds when compared with the AES.
*Security* The security of LED is studied in two papers. In [24], the authors investigate the security of LED against the meet-in-the-middle attacks. They are able to mount meet-in-the-middle attacks against 8 rounds of LED-64 and 16 rounds of LED-128 with complexities slightly lower than the exhaustive key search. In [41], the authors present results concerning differential cryptanalysis of LED. They first show attacks for LED-64 reduced to 12 and 16 rounds and finally present an observation on full LED in the related-key settings.
*Implementation* We made three implementations of this block cipher. The first one (LEDxx) is a standard and implementation that does not make use tables. The others use 8 lookup-tables in the `SubCells`, `ShiftRows` and `MixColumnsSerial`. In one of our implementations (LEDxx_tdur), these tables are pre-computed and we don't evaluate the cost of building them. In the other, these tables are computed before encrypting and decrypting. To store the block to cipher and the key, we use arrays of 8-bit numbers.

***mCrypton*** mCrypton [39] is the lightweight version of the block cipher Crypton [38]. mCrypton is a 64 bits block cipher with three possible key lengths 64, 96 or 128 bits. It uses a SPN structure repeated during 12 rounds that acts on a 4 × 4 nibbles matrix. The round function uses four elementary transformations: a Sbox layer, a bit permutation, a column-to-row transposition and a subkey addition.
*Security* In [45, 40], the security of mCrypton is scrutinized in the related key settings. In [45], the author shows that 8-round mCrypton with 128-bit key is vulnerable to related-key rectangle attack. In [40], the authors construct 9-round related-key impossible differential attacks against mCrypton-96 and mCrypton-128. No result appears in the unknown key settings.
*Implementation* We implemented the solution proposed in the reference paper, using a 8-bit array for the block to cipher, and a 16-bit table for the key.

***MIBS*** MIBS [25] uses a Feistel structure with data block length of 64 bits and key lengths of 64 bits or 80 bits and consists of 32 rounds. The internal F-function, inspired from the one of the Camellia block cipher [1], acts on nibbles and is composed of a subkey addition, an Sbox layer, a linear mixing layer and a nibble-wise permutation.
*Security* In [3], the authors present linear attacks on up to 18-round MIBS, the first ciphertext-only attacks on 13-round MIBS, a differential analysis on 14-round MIBS, and an impossible differential on 12-round MIBS. These attacks do not threaten the full 32-round MIBS, but significantly reduce its security margin.
*Implementation* We used a table for the sbox function and an other for the inverse. We store the block to cipher and the key on tables of 8-bit numbers.

***Noekeon*** Noekeon [14] is a conventional block cipher with a block length and a key length of 128 bits submitted to the Nessie project. It is a substitution-linear transformation network in bit-slice

mode that allows very fast and compact implementations. It is similar to Serpent and uses cyclic shifts and bit-wise Boolean operations followed by an Sbox layer that acts on nibbles. It is composed of 16 rounds followed by a simple output transformation. Noekeon has two key schedules, one for applications where related-key attacks are not considered dangerous and one for applications where related-key attacks can be mounted.
*Security* Noekeon has not been chosen by the Nessie project due to the analysis done by Knudsen and Raddum in [32]. They show that there exist many related keys for which plaintexts of certain differences result in ciphertexts of certain differences with high probabilities independent of the key schedule used. It is also shown that for six out of seven S-boxes which satisfy the design criteria of the Noekeon designers, the resulting block ciphers are vulnerable to either a differential attack, a linear attack or both. It is concluded that Noekeon is not designed according to an optimal diffusion strategy.
*Implementation* We implemented two variants of the algorithm. In the first one (*INDNoekeon* in the following), a key scheduling phase is applied. In the second one (*DIRNoekeon*), this phase is skipped and we use the cipher itself in replacement. The block to cipher and the key are stored in tables of 32-bit entries.

***Piccolo*** Piccolo [47] is a 64 bits blockcipher supporting 80 and 128 bits keys. It mixes together a 4 branches Feistel structure followed by a byte permutation $RP$. The two F-functions that are called in the Feistel layer act on 16 bits words and are composed of a Sbox layer applied at nibble level and of a nibble MixColumns like operation followed by a subkey addition. Piccolo is one of the first block cipher that has an hardware implementation requiring less than 1000 gates.
*Security* All the results concerning the security of Piccolo focus on biclique cryptanalysis. The best result in this direction is presented in [26] where bicliques on full round versions of Piccolo-80 and Piccolo-128 slightly lower than exhaustive key search are described.
*Implementation* We used one table storing the results of the multiplication of 0..15 by 2, and an other for the multiplication by 3. We store the block to cipher and the key in tables of 16-bit numbers.

***PRESENT*** PRESENT is the most famous lightweight block cipher presented at CHES 2007 [8]. It ciphers block of length 64 bits under keys of lengths 80 or 128 bits. The number of rounds is equal to 31. The round function is a simple SPN network composed of a subkey addition, a Sbox layer calling always the same nibble Sbox and a bit permutation layer.
*Security* PRESENT has attracted a lot of cryptanalytic attention because of very particular linear biases. The papers [42, 12, 13, 36] study the linear behavior of PRESENT regarding multiple linear trails. This kind of cryptanalysis allows to mount multi-linear attacks on up to 27 rounds of PRESENT but using all the codebook. Two bicliques with complexities about the same than the ones of the exhaustive key search against the two versions of PRESENT are also proposed in [26].
*Implementation* We store the block to cipher and the key in tables of 16-bit numbers.

***SEA*** SEA-$n, b$ [50] is a lightweight block cipher that takes an $n$ bits block under a key of length also $n$. It acts on words of size $b$ bits and has $nr$ rounds. It is a very suitable block cipher as $n$ could take the values 48, 96, 144, etc. It is based on a modified two branches Feistel network. The F-function of the Feistel is constructed using elementary operations and is composed of an addition with the subkey, an Sbox layer that acts on $b$ bits word and words and bits rotations. The recommended number of rounds $nr$ is equal to $3n/4 + n/b + 2 * \left\lfloor \frac{b}{2} \right\rfloor$.
*Security* Up to our knowledge, there is no security analysis published about SEA except the ones included in the original paper [50].
*Implementation* To store the block to cipher and the key, we use arrays of 16-bit numbers.

***SKIPJACK*** SKIPJACK [44] is a block cipher developed by the NSA and declassified in 1998. SKIPJACK uses a 80 bits key and 64 bits data blocks. It is an unbalanced Feistel network with 32 rounds. It has two types of rounds, called Rule A and Rule B. Each round is described as a

linear feedback shift register with an additional nonlinear keyed $G$ permutation. Rule B is basically the inverse of Rule A with minor positioning differences. Skipjack applies eight rounds of Rule A, followed by eight rounds of Rule B, followed by another eight rounds of Rule A, followed by another eight rounds of Rule B. $G$ is a four-round Feistel permutation composed of an 8 bits Sbox and an 8 bits subkey addition.
*Security* SKIPJACK has been subject to intensive analysis as summed-up in [29]. The currently most successful attack against the cipher is the impossible differential attack which breaks 31 rounds out of 32, marginally faster than exhaustive search [4].
*Implementation* We store the block to cipher and the key in tables of 8-bit numbers.

***TEA and XTEA*** TEA (Tiny Encryption Algorithm) is an old block cipher notable for its simplicity. It was designed in 1994 by D. Wheeler and R. Needham [56]. Due to many weaknesses found against TEA (see for example [27] for more details), TEA has been replaced by XTEA in [43]. XTEA is a 64 bits block cipher with a 128 bits key. It is based on a Feistel network and the recommended number of rounds is 64. The internal F-function is really simple and is composed of left and right shifts, XORs and additions.
*Security* Many papers have analyzed the security of XTEA. We will focus here on the most recent publications. In [11], the authors present an impossible differential attacks on 23-round XTEA. In [46], a three-subset meet-in-the-middle attack is applied against 25 rounds of XTEA with 9 known plaintexts and $2^{120.4}$ XTEA computations.
*Implementation* We store the block to cipher and the key in tables of 32-bit numbers.

***TWINE*** TWINE is a lightweight 64 bits block cipher [51] having 80 bits or 128 bits key. It employs a Generalized Feistel Structure with 16 branches. It has 36 rounds whatever the key length. The internal F-function, repeated 8 times per round, is just composed of a subkey addition and of a single Sbox that acts on nibbles.
*Security* In [10], the authors present two biclique attacks on TWINE-80 and TWINE-128 with time complexities equal to $2^{79.10}$ and $2^{126.82}$ respectively with a data requirement for the two attacks equal to $2^{60}$.
*Implementation* We store the subkeys and the block to cipher in tables of 8-bit numbers.

***Conclusion*** In conclusion, we could notice that all the studied block ciphers have a sufficient security margin to be employed in real life applications. The most risky ones seem to be KLEIN, Noekeon and SKIPJACK.

## 2 Methodology

In this section, we present the platform used to perform the benchmarks and we also describe the testing framework.

### 2.1 The dedicated platform

The MSP430 is a Texas Instrument microcontroller running with an external 8MHz clock. This microcontroller is programmable via a JTAG connection. It integrates a 48 KBytes flash memory, a 10 Koctets RAM memory, 48 configurable Inputs/Outputs, 12-bit analog-to-digital conversion pins, a watchdog, 2 serial communication ports and 2 configurable timers. This microcontroller is compatible with most of real-time operating systems such as FreeRTOS.

All the codes were written in C. We used the GCC toolchain for MSP430 family to flash programs into the microcontroller. This includes the GNU C compiler (GCC), the assembler and linker (binutils), the debugger (GDB), and some other tools needed to make a complete development environment for the MSP430. These tools can be used on Windows, Linux, BSD and most other flavors of Unix. We used msp430-gcc version 4.6.3.

### 2.2 Methodology

We measured the performance of the algorithms as well as the memory consumption. To obtain the performance, we used simulator coming with mspdebug. This simulator is able to give the number of clock cycles spent at any point of the program execution. Although it is only a simulator, it is cycle-accurate and the experiments we made on real hardware confirmed the results obtained.

Concerning the memory consumption, we distinguish between the need of read-only memory (ROM) and writable memory (RAM). The ROM is used to store the code as well as tables that do not need to be modified – for instance, the F-table of skipjack. We obtain the size of the ROM needed simply by declaring as *static const* the concerned variables and getting the size of the text section in the *elf* file. In order to get the size of RAM needed, mspdebug tells us until which address the execution stack was modified.

## 3 Results

### 3.1 CPU cycles and energy consumption

Table 2 gives the performance of the algorithms.

### 3.2 Memory requirements

Table 2 reports the memory consumption of the algorithms. It shows the requirements of read-only memory (code + read-only tables) as well as the amount of RAM needed to store the stack and modifiable data. We can see that the requirement of RAM is very similar and very small, except for the CLEFIA and the KATAN families. The memory requirements of these functions is due to the use of large tables in the key scheduling phase.

On the contrary, the need of read-only memory is very different from one algorithm to an other. Whereas TEA and XTEA requires only 1354 and 1394 bytes of ROM to execute, KTANTAN requires 16252 bytes in its 64-bits version. The ROM consumption of the KATAN family is due to the tables used to store the bitfields (see Section 1).

### 3.3 Analysis

We consider 6 different metrics here: cycle count for enc.+key and for dec.+key, cycles/bytes for enc.+key and for dec.+key, code size (in bytes), RAM use (in bytes) and the metric introduced in [17] that is code size × cycle count product, normalized by the block size (see Fig. 1). We detail in this Section some particular observations.

First, due to sensor memory requirement, we consider compact implementations. As shown in Table 3, TEA ans XTEA have memory size less than 1500 bytes whereas NOEKEON, LED mCrypton, Piccolo, SEA and TWINE have memory footprint between 2000 and 3000 bytes which is really reasonable. At the contrary, all the KATAN and KTANTAN version have huge memory footprints due to their particular design which has the same cost when enciphering/deciphering 32, 48 or 64 blocks in parallel. In terms of RAM occupancy, HIGHT, LBlock, mCrypton, MIBS, Skipjack, TEA and XTEA require less than 20 bytes of RAM which is really performing.

Concerning performance, TEA, XTEA and the AES are the only ones that require less than 2000 cycles/byte. Some lightweight designs have poor performance: KATAN, KTANTAN, LED, mCrypton and PRESENT whereas the others (DESXL, NOEKEON, HIGHT, KLEIN, LBlock, Piccolo, TWINE) use less than 5500 cycles/bytes. IDEA is efficient in encryption but as expected and due to the key schedule inefficient in decryption.

Lastly, the combined metric in Figure 1 first shows the excellent size vs. performance trade-off offered by the AES. Among the low-cost ciphers, NOEKEON, TEA and XTEA have also an excellent behavior. In the same way, HIGHT, Piccolo and TWINE provide good trade-offs whereas KATAN and KTANTAN are not present in the Figure due to their too bad behaviors.

| Algorithm | Block Size | Enc.+key: | Enc.+key : | Dec.+key: | Dec.+key: cycles/byte |
|---|---|---|---|---|---|
| | (bits) | cycle count | cycles/byte | cycle Count | |
| AES | 128 | 30257 | 1891 | 38508 | 2406 |
| CLEFIA128 | 128 | 98145 | 6134 | 101855 | 6365 |
| CLEFIA192 | 128 | 150314 | 9394 | 123333 | 7708 |
| CLEFIA256 | 128 | 155658 | 9728 | 145291 | 9080 |
| DESXL | 64 | 26055 | 3256 | 66913 | 8364 |
| DIRnoekeon | 128 | 26291 | 1643 | 27129 | 1695 |
| HIGHT | 64 | 32372 | 4046 | 32623 | 4077 |
| IDEA | 64 | 31402 | 3925 | 163380 | 20422 |
| INDnoekeon | 128 | 52564 | 3285 | 53435 | 3339 |
| KATAN32 | 32 | 744279 | 186069 | 717056 | 179264 |
| KATAN48 | 48 | 1127271 | 187878 | 1053680 | 175613 |
| KATAN64 | 64 | 1518391 | 189798 | 1397924 | 174740 |
| KLEIN64 | 64 | 29514 | 3689 | 100600 | 12575 |
| KLEIN80 | 64 | 40278 | 5034 | 135369 | 16921 |
| KLEIN96 | 64 | 51502 | 6437 | 170789 | 21348 |
| KTANTAN32 | 32 | 10233211 | 2558302 | 10193489 | 2548372 |
| KTANTAN48 | 48 | 10614933 | 1769155 | 10525067 | 1754177 |
| KTANTAN64 | 64 | 11004783 | 1375597 | 10864265 | 1358033 |
| LBlock | 64 | 42954 | 5369 | 22005 | 2750 |
| LED128 | 64 | 1341488 | 167686 | 1345152 | 168144 |
| LED128_tcalc | 64 | 268721 | 33590 | 274953 | 34369 |
| LED128_tdur | 64 | 171056 | 21382 | 173832 | 21729 |
| LED64 | 64 | 894680 | 111835 | 897352 | 112169 |
| LED64_tcalc | 64 | 212409 | 26551 | 217401 | 27175 |
| LED64_tdur | 64 | 114872 | 14359 | 116280 | 14535 |
| MCRYPTON64 | 64 | 107803 | 13475 | 219870 | 27483 |
| MCRYPTON96 | 64 | 108499 | 13562 | 220320 | 27540 |
| MCRYPTON128 | 64 | 108415 | 13551 | 220568 | 27571 |
| MIBS64 | 64 | 49056 | 6132 | 52890 | 6611 |
| MIBS80 | 64 | 58688 | 7336 | 39842 | 4980 |
| PRESENT_SIZE | 64 | 491602 | 61450 | 489813 | 61226 |
| PRESENT_SPEED | 64 | 364587 | 45573 | 368731 | 46091 |
| Piccolo128 | 64 | 36497 | 4562 | 39600 | 4950 |
| Piccolo80 | 64 | 32106 | 4013 | 34630 | 4328 |
| SEA | 96 | 119455 | 9954 | 120158 | 10013 |
| SKIPJACK | 64 | 84923 | 10615 | 123368 | 15421 |
| TEA | 64 | 8785 | 1098 | 9129 | 1141 |
| TWINE | 128 | 82003 | 5125 | 60932 | 3808 |
| XTEA | 64 | 9287 | 1160 | 9631 | 1203 |

**Table 2.** Software performance.

| Function | RAM requirement (bytes) | Size of read-only data (bytes) |
|---|---|---|
| AES | 19 | 4460 |
| CLEFIA128 | 180 | 4780 |
| CLEFIA192 | 268 | 5010 |
| CLEFIA256 | 268 | 4924 |
| DESXL | 112 | 16816 |
| DIRnoekeon | 34 | 2710 |
| HIGHT | 18 | 3130 |
| IDEA | 82 | 3140 |
| INDnoekeon | 34 | 2784 |
| KATAN32 | 1881 | 5816 |
| KATAN48 | 1969 | 7076 |
| KATAN64 | 1953 | 8348 |
| KLEIN64 | 36 | 5486 |
| KLEIN80 | 38 | 5676 |
| KLEIN96 | 39 | 5862 |
| KTANTAN32 | 614 | 10516 |
| KTANTAN48 | 702 | 11764 |
| KTANTAN64 | 790 | 16252 |
| LBlock | 13 | 3568 |
| LED128 | 41 | 2648 |
| LED128_tcalc | 41 | 2948 |
| LED128_tdur | 41 | 2264 |
| LED64 | 41 | 2670 |
| LED64_tcalc | 41 | 2498 |
| LED64_tdur | 41 | 2264 |
| MCRYPTON64 | 18 | 2726 |
| MCRYPTON96 | 20 | 2834 |
| MCRYPTON128 | 24 | 3108 |
| MIBS64 | 29 | 3184 |
| MIBS80 | 16 | 3138 |
| PRESENT_SIZE | 142 | 4964 |
| PRESENT_SPEED | 142 | 4814 |
| Piccolo128 | 91 | 2510 |
| Piccolo80 | 79 | 2434 |
| SEA | 24 | 2804 |
| SKIPJACK | 19 | 6628 |
| TEA | 13 | 1354 |
| TWINE | 23 | 2216 |
| XTEA | 11 | 1394 |

**Table 3.** Memory usage.

## 4 Conclusion

We have presented here some benchmarks performed on lightweight block ciphers, the traditional ones and the new ones on a dedicated platform which is a sensor. In total, 17 ciphers have been implemented and analyzed keeping in mind that the compactness is an important issue in the sensor world. They show that some well-suited block ciphers such as Piccolo, TWINE, XTEA or the AES have good performance considering the trade-off between code size and cycle count. We also see that most of the ciphers specially dedicated to hardware (such as LED, PRESENT or KATAN and KTANTAN) have poor results.

## References

1. Kazumaro Aoki, Tetsuya Ichikawa, Masayuki Kanda, Mitsuru Matsui, Shiho Moriai, Junko Nakajima, and Toshio Tokita. Camellia: A 128-bit block cipher suitable for multiple platforms - design and analysis. In *Selected Areas in Cryptography - SAC 2000*, volume 2012 of *Lecture Notes in Computer Science*, pages 39–56. Springer, 2000.
2. Jean-Philippe Aumasson, María Naya-Plasencia, and Markku-Juhani O. Saarinen. Practical attack on 8 rounds of the lightweight block cipher klein. In *Progress in Cryptology - INDOCRYPT 2011*, volume 7107 of *LNCS*, pages 134–145. Springer, 2011.

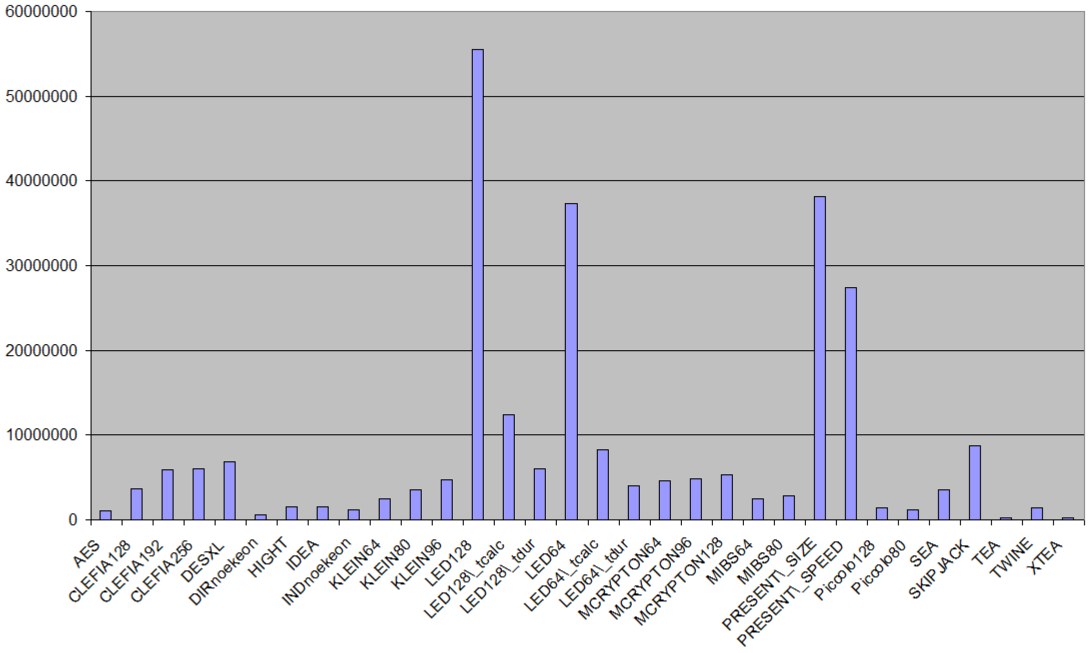

**Fig. 1.** Metric introduced in [17]: code size × cycle count product/block size.

3. Asli Bay, Jorge Nakahara, and Serge Vaudenay. Cryptanalysis of reduced-round mibs block cipher. In *Cryptology and Network Security - CANS 2010*, volume 6467 of *LNCS*, pages 1–19. Springer, 2010.
4. Eli Biham, Alex Biryukov, and Adi Shamir. Cryptanalysis of skipjack reduced to 31 rounds using impossible differentials. In *Advances in Cryptology - EUROCRYPT '99*, volume 1592 of *Lecture Notes in Computer Science*. Springer, 1999.
5. Eli Biham, Orr Dunkelman, and Nathan Keller. A new attack on 6-round idea. In *Fast Software Encryption - FSE 2007*, volume 4593 of *Lecture Notes in Computer Science*, pages 211–224. Springer, 2007.
6. Alex Biryukov, Jorge Nakahara Jr., Bart Preneel, and Joos Vandewalle. New weak-key classes of idea. In *ICICS*, volume 2513 of *Lecture Notes in Computer Science*, pages 315–326. Springer, 2002.
7. Andrey Bogdanov, Dmitry Khovratovich, and Christian Rechberger. Biclique cryptanalysis of the full aes. In *Advances in Cryptology - ASIACRYPT 2011*, volume 7073 of *Lecture Notes in Computer Science*, pages 344–371. Springer, 2011.
8. Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, Christof Paar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe. PRESENT: An Ultra-Lightweight Block Cipher. In *Cryptographic Hardware and Embedded Systems - CHES 2007*, LNCS 4727, pages 450–466. Springer, 2007.
9. Christophe De Cannière, Orr Dunkelman, and Miroslav Knezevic. Katan and ktantan - a family of small and efficient hardware-oriented block ciphers. In *Cryptographic Hardware and Embedded Systems - CHES 2009*, volume 5747 of *Lecture Notes in Computer Science*, pages 272–288. Springer, 2009.
10. Mustafa Çoban, Ferhat Karakoç, and Özkan Boztaş. Biclique cryptanalysis of twine. Cryptology ePrint Archive, Report 2012/422, 2012. `http://eprint.iacr.org/`.
11. Jiazhe Chen, Meiqin Wang, and Bart Preneel. Impossible Differential Cryptanalysis of the Lightweight Block Ciphers TEA, XTEA and HIGHT. In *Progress in Cryptology - AFRICACRYPT 2012*, volume 7374 of *Lecture Notes in Computer Science*, pages 117–137. Springer, 2012.
12. Baudoin Collard and François-Xavier Standaert. A Statistical Saturation Attack against the Block Cipher PRESENT. In *Topics in Cryptology - CT-RSA 2009*, LNCS 5473, pages 195–210. Springer, 2009.
13. Baudoin Collard and François-Xavier Standaert. Multi-trail statistical saturation attacks. In *Applied Cryptography and Network Security - ACNS 2010*, volume 6123 of *Lecture Notes in Computer Science*, pages 123–138. Springer, 2010.
14. Joan Daemen, Michaël Peeters, Gilles Van Assche, and Vincent Rijmen. Nessie proposal: Noekeon. Submitted as an NESSIE Candidate Algorithm, 2000. `http://gro.noekeon.org/`.
15. Patrick Derbez, Pierre-Alain Fouque, and Jérémy Jean. Improved key recovery attacks on reduced-round aes in the single-key setting. Cryptology ePrint Archive, Report 2012/477, 2012. `http://eprint.iacr.org/`.
16. Clément Burin des Roziers, Guillaume Chelius, Tony Ducrocq, Eric Fleury, Antoine Fraboulet, Antoine Gallais, Nathalie Mitton, Thomas Noël, and Julien Vandaele. Using senslab as a first class scientific tool for large scale wireless sensor network experiments. In *NETWORKING 2011*, volume 6640 of *Lecture Notes in Computer Science*, pages 147–159. Springer, 2011.
17. Thomas Eisenbarth, Zheng Gong, Tim Güneysu, Stefan Heyse, Sebastiaan Indesteege, Stéphanie Kerckhof, François Koeune, Tomislav Nad, Thomas Plos, Francesco Regazzoni, François-Xavier Standaert, and Loïc van

Oldeneel tot Oldenzeel. Compact implementation and performance evaluation of block ciphers in attiny devices. In *Progress in Cryptology - AFRICACRYPT 2012*, volume 7374 of *Lecture Notes in Computer Science*, pages 172–187. Springer, 2012.

18. FIPS 197. Advanced Encryption Standard. Federal Information Processing Standards Publication 197, 2001. U.S. Department of Commerce/N.I.S.T.
19. Zheng Gong, Svetla Nikova, and Yee Wei Law. Klein: A new family of lightweight block ciphers. In *RFID. Security and Privacy - RFIDSec 2011*, volume 7055 of *Lecture Notes in Computer Science*, pages 1–18. Springer, 2011.
20. Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw. The led block cipher. In *Cryptographic Hardware and Embedded Systems - CHES 2011*, volume to appear of *LNCS*. Springer, 2011.
21. Philip Hawkes. Differential-linear weak key classes of idea. In *Advances in Cryptology - EUROCRYPT '98*, volume 1403 of *Lecture Notes in Computer Science*, pages 112–126. Springer, 1998.
22. Deukjo Hong, Bonwook Koo, and Daesung Kwon. Biclique attack on the full hight. In *Information Security and Cryptology - ICISC 2011*, volume 7259 of *Lecture Notes in Computer Science*, pages 365–374. Springer, 2011.
23. Deukjo Hong, Jaechul Sung, Seokhie Hong, Jongin Lim, Sangjin Lee, Bonseok Koo, Changhoon Lee, Donghoon Chang, Jaesang Lee, Kitae Jeong, Hyun Kim, Jongsung Kim, and Seongtaek Chee. HIGHT: A New Block Cipher Suitable for Low-Resource Device. In *Cryptographic Hardware and Embedded Systems - CHES 2006*, LNCS 4249, pages 46–59. Springer, 2006.
24. Takanori Isobe and Kyoji Shibutani. Security analysis of the lightweight block ciphers xtea, led and piccolo. In *Information Security and Privacy - ACISP 2012*, volume 7372 of *Lecture Notes in Computer Science*, pages 71–86. Springer, 2012.
25. Maryam Izadi, Babak Sadeghiyan, Seyed Saeed Sadeghian, and Hossein Arabnezhad Khanooki. MIBS: A New Lightweight Block Cipher. In *Cryptology and Network Security - CANS 2009*, LNCS 5888, pages 334–348, 2009.
26. Kitae Jeong, HyungChul Kang, Changhoon Lee, Jaechul Sung, and Seokhie Hong. Biclique cryptanalysis of lightweight block ciphers present, piccolo and led. Cryptology ePrint Archive, Report 2012/621, 2012. `http://eprint.iacr.org/`.
27. John Kelsey, Bruce Schneier, and David Wagner. Related-key cryptanalysis of 3-way, biham-des, cast, des-x, newdes, rc2, and tea. In *Information and Communication Security - ICICS'97*, volume 1334 of *Lecture Notes in Computer Science*, pages 233–246. Springer, 1997.
28. Dmitry Khovratovich, Gaëtan Leurent, and Christian Rechberger. Narrow-bicliques: Cryptanalysis of full idea. In *Advances in Cryptology - EUROCRYPT 2012*, volume 7237 of *Lecture Notes in Computer Science*, pages 392–410. Springer, 2012.
29. Jongsung Kim and Raphael C.-W. Phan. A cryptanalytic view of the nsa's skipjack block cipher design. In *Advances in Information Security and Assurance - ISA 2009*, volume 5576 of *Lecture Notes in Computer Science*, pages 368–381. Springer, 2009.
30. Simon Knellwolf, Willi Meier, and María Naya-Plasencia. Conditional differential cryptanalysis of nlfsr-based cryptosystems. In *Advances in Cryptology - ASIACRYPT 2010*, volume 6477 of *Lecture Notes in Computer Science*, pages 130–145. Springer, 2010.
31. Simon Knellwolf, Willi Meier, and María Naya-Plasencia. Conditional differential cryptanalysis of trivium and katan. In *Selected Areas in Cryptography - SAC 2011*, volume 7118 of *Lecture Notes in Computer Science*, pages 200–212. Springer, 2011.
32. Lars R. Knudsen and Havard Raddum. On noekeon, 2001. `https://www.cosic.esat.kuleuven.be/nessie/reports/phase1/uibwp3-009.pdf`.
33. Bonwook Koo, Deukjo Hong, and Daesung Kwon. Related-key attack on the full hight. In *Information Security and Cryptology - ICISC 2010*, volume 6829 of *Lecture Notes in Computer Science*, pages 49–67. Springer, 2010.
34. Xuejia Lai and James L. Massey. A proposal for a new block encryption standard. In *Advances in Cryptology - EUROCRYPT '90*, volume 473 of *Lecture Notes in Computer Science*, pages 389–404. Springer, 1990.
35. Yee Wei Law, Jeroen Doumen, and Pieter H. Hartel. Survey and benchmark of block ciphers for wireless sensor networks. *TOSN*, 2(1):65–93, 2006.
36. Gregor Leander. On linear hulls, statistical saturation attacks, present and a cryptanalysis of puffin. In *Advances in Cryptology - EUROCRYPT 2011*, volume 6632 of *Lecture Notes in Computer Science*, pages 303–322. Springer, 2011.
37. Gregor Leander, Christof Paar, Axel Poschmann, and Kai Schramm. New lightweight des variants. In *Fast Software Encryption - FSE 2007*, volume 4593 of *Lecture Notes in Computer Science*, pages 196–210. Springer, 2007.
38. Chae Hoon Lim. A revised version of crypton - crypton v1.0. In *Fast Software Encryption - FSE'99*, volume 1636 of *Lecture Notes in Computer Science*, pages 31–45. Springer, 1999.
39. Chae Hoon Lim and Tymur Korkishko. mCrypton - A Lightweight Block Cipher for Security of Low-Cost RFID Tags and Sensors. In *Workshop on Information Security Applications - WISA 2005*, LNCS 3786, pages 243–258. Springer Verlag, 2005.
40. Hamid Mala, Mohammad Dakhilalian, and Mohsen Shakiba. Cryptanalysis of mcrypton - a lightweight block cipher for security of rfid tags and sensors. *Int. J. Communication Systems*, 25(4):415–426, 2012.
41. Florian Mendel, Vincent Rijmen, Deniz Toz, and Kerem Varici. Differential analysis of the led block cipher. In *Advances in Cryptology - ASIACRYPT 2012*, volume 7658 of *Lecture Notes in Computer Science*, pages 190–207. Springer, 2012.

42. Jorge Nakahara, Pouyan Sepehrdad, Bingsheng Zhang, and Meiqin Wang. Linear (Hull) and Algebraic Cryptanalysis of the Block Cipher PRESENT. In *Cryptology and Network Security - CANS 2009*, LNCS 5888, pages 58–75. Springer, 2009.
43. Roger M. Needham and David J. Wheeler. Tea extensions. Technical report, Computer Laboratory, University of Cambridge, October 1997.
44. NIST. Skipjack and kea algorithm specification. Technical Report, 1998. `http://csrc.nist.gov/groups/STM/cavp/documents/skipjack/skipjack.pdf`.
45. Jong Hyuk Park. Security analysis of mcrypton proper to low-cost ubiquitous computing devices and applications. *Int. J. Communication Systems*, 22(8):959–969, 2009.
46. Yu Sasaki, Lei Wang, Yasuhide Sakai, Kazuo Sakiyama, and Kazuo Ohta. Three-subset meet-in-the-middle attack on reduced xtea. In *Progress in Cryptology - AFRICACRYPT 2012*, volume 7374 of *Lecture Notes in Computer Science*, pages 138–154. Springer, 2012.
47. Kyoji Shibutani, Takanori Isobe, Harunaga Hiwatari, Atsushi Mitsuda, Toru Akishita, and Taizo Shirai. Piccolo: An ultra-lightweight blockcipher. In *Cryptographic Hardware and Embedded Systems - CHES 2011*, volume 6917 of *LNCS*. Springer, 2011.
48. Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai, and Tetsu Iwata. The 128-bit blockcipher clefia (extended abstract). In *Fast Software Encryption - FSE 2007*, volume 4593 of *Lecture Notes in Computer Science*, pages 181–195. Springer, 2007.
49. Hadi Soleimany and Kaisa Nyberg. Zero-correlation linear cryptanalysis of reduced-round lblock. Cryptology ePrint Archive, Report 2012/570, 2012. `http://eprint.iacr.org/`.
50. François-Xavier Standaert, Gilles Piret, Neil Gershenfeld, and Jean-Jacques Quisquater. Sea: A scalable encryption algorithm for small embedded applications. In *Smart Card Research and Advanced Applications - CARDIS 2006*, volume 3928 of *Lecture Notes in Computer Science*, pages 222–236. Springer, 2006.
51. Tomoyasu Suzaki, Kazuhiko Minematsu, Sumio Morioka, and Eita Kobayashi. Twine: A lightweight block cipher for multiple platforms. In *Selected Areas in Cryptography - SAC 2012*, volume to appear of *Lecture Notes in Computer Science*. Springer, 2012.
52. Cihangir Tezcan. The improbable differential attack: Cryptanalysis of reduced round clefia. In *Progress in Cryptology - INDOCRYPT 2010*, volume 6498 of *Lecture Notes in Computer Science*, pages 197–209. Springer, 2010.
53. Yukiyasu Tsunoo, Etsuko Tsujihara, Maki Shigeri, Teruo Saito, Tomoyasu Suzaki, and Hiroyasu Kubo. Impossible differential cryptanalysis of clefia. In *Fast Software Encryption - FSE 2008*, volume 5086 of *Lecture Notes in Computer Science*, pages 398–411. Springer, 2008.
54. Yanfeng Wang, Wenling Wu, Xiaoli Yu, and Lei Zhang. Security on lblock against biclique cryptanalysis. In *Information Security Applications - WISA 2012*, volume 7690 of *Lecture Notes in Computer Science*, pages 1–14. Springer, 2012.
55. Lei Wei, Christian Rechberger, Jian Guo, Hongjun Wu, Huaxiong Wang, and San Ling. Improved meet-in-the-middle cryptanalysis of ktantan (poster). In *Information Security and Privacy - ACISP 2011*, volume 6812 of *Lecture Notes in Computer Science*, pages 433–438. Springer, 2011.
56. David J. Wheeler and Roger M. Needham. Tea, a tiny encryption algorithm. In *Fast Software Encryption - FSE 94*, volume 1008 of *LNCS*, pages 363–366. Springer, 1994.
57. Wenling Wu and Lei Zhang. Lblock: A lightweight block cipher. In *Applied Cryptography and Network Security - ACNS 2011*, volume 6715 of *LNCS*, pages 327–344. Springer, 2011.