Pei *et al. EURASIP Journal on Wireless Communications and Networking* (2018) 2018:117
https://doi.org/10.1186/s13638-018-1121-6

EURASIP Journal on Wireless Communications and Networking

RESEARCH Open Access

Trade-off of security and performance of lightweight block ciphers in Industrial Wireless Sensor Networks

Chao Pei[1,2,3], Yang Xiao[4*], Wei Liang[1,2*] and Xiaojia Han[1,2,3]

Abstract

Lightweight block ciphers play an indispensable role for the security in the context of pervasive computing. However, the performance of resource-constrained devices can be affected dynamically by the selection of suitable cryptalgorithms, especially for the devices in the resource-constrained devices and/or wireless networks. Thus, in this paper, we study the trade-off between security and performance of several recent top performing lightweight block ciphers for the demand of resource-constrained Industrial Wireless Sensor Networks. Then, the software performance evaluation about these ciphers has been carried out in terms of memory occupation, cycles per byte, throughput, and a relative good comprehensive metric. Moreover, the results of avalanche effect, which shows the possibility to resist possible types of different attacks, are presented subsequently. Our results show that SPECK is the software-oriented lightweight cipher which achieves the best performance in various aspects, and it enjoys a healthy security margin at the same time. Furthermore, PRESENT, which is usually used as a benchmark for newer hardware-oriented lightweight ciphers, shows that the software performance combined with avalanche effect is inadequate when it is implemented. In the real application, there is a need to better understand the resources of dedicated platforms and security requirement, as well as the emphasis and focus. Therefore, this case study can serve as a good reference for the better selection of trade-off between performance and security in constrained environments.

1 Introduction

In the traditional resource-constrained environment, the constrained devices such as nodes in wireless sensor networks and radio-frequency identification (RFID) tags usually have the characteristics of weak computation ability, extremely small storage space, and strictly usage of power consumption [1–3]. Especially in the context of Internet of Things (IOTs), small embedded devices with poor computing capability are expected to connect to larger networks [4–7]. Although great changes and developments are brought to our society and life, it is the fact that almost all of the applications are inevitably faced with potential threats of information security [8]. As increasingly sensitive information is transmitted and manipulated, cryptographic protection should be made. Wearable devices, medical sensing networks, or the sensor networks for the military surveillance are such examples that security about them should pay more attentions [9–12]. Especially in the complex industry environment, interference such as high humidity, high vibration, variable temperature, and multi-frequency noise always exists. Hence, considering devices with constrained resources combined with sufficient security, there is no doubt that the performance about these environments are difficult to guarantee perfectly.

Meanwhile, the term "lightweight" is frequently used and mentioned in many literatures [13], but there is not an accurate definition about it. Ciphers targeted for resource-constrained devices are regarded as lightweight ciphers, and either software or hardware implementations should improve the utilization rate of resource. What is shown in [14] is that operations such as block sizes, key sizes, and the process of key scheduling should take into consideration. Elementary operations such as addition,

*Correspondence: yangxiao@ieee.org; weiliang@sia.cn
[4]Department of Computer Science, The University of Alabama, Tuscaloosa, 5487-02903 AL, USA
[1]Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, 110016, Shenyang, China
Full list of author information is available at the end of the article

AND, OR, exclusive, or shift are welcomed because simple operations can be applied to all elementary platforms. Moreover, small blocks and short-key length in some means can simplify the encryption process. Of course, resource-constrained environments between wireless sensor networks (WSNs) and RFID are quite different. As for WSNs, sensor nodes with microcontrollers are grouped with sensing units, storage units, transceiver, and other components, and the size of different hardware platforms changes in a large range. In addition, sensor nodes are almost battery powered; energy efficiency and extensive mote life span are expected. Thus, choosing a cipher that could match the resources of nodes is an important consideration [15]. As for RFID, the most transponders are passive RFID tags, and there are different kinds of devices with different requirements, prices, and usage with different capabilities [16]. The electronic product code tags, in which the ultra-high-frequency is adopted at the band of 860–960 MHz, are usually used, and the price of each tag is approximately 0.15 USD. One of the early lightweight cryptographic attempts for RFID includes the work [17], in which they claim that the hardware implementations about security portion should be under 2000 gate equivalents (GEs). An ISO/IEC standard on lightweight cryptography stated that the design requirements should be made with 1000–2000 GEs [18]. The paper [19] claims that a total number of about 1000 to 10000 gates are included in an RFID tag, and just with 200 to 2000 GEs will be available specifically for security, and the standardizing cryptographic such as Advanced Encryption Standard (AES) is not suitable. Of course, there are many other literatures to investigate the resources available on tags, and these detail information could be found in [20, 21].

Because of the requirement and the fact that ciphers are the backbone of data protection and secrecy for highly sensitive and classified data, as a result, many lightweight block ciphers have been proposed in order to allow strong security guarantees at a low cost for these resource-constrained environments [15, 22–24]. Block ciphers with limited to small GEs could have the possibility to satisfy lightweight environments and real-time applications. For security, the total GEs available should be approximately 2000–3000. Extensions to Tiny Encryption Algorithm (XTEA) and Corrected Block Tiny Encryption Algorithm (XXTEA) [25] are designed to deal with the weakness of tiny encryption algorithm (TEA), which is a tiny but fast block cipher. However, there is no much information about hardware implementation results, and XTEA is vulnerable to a related key differential attack and a related rectangle attack, while at the same time, XXTEA suffers from a chosen plaintext attack [26]. At 2007, Data Encryption Standard Lightweight (DESL) and XORed variant of DESL (DESXL), lightweight variants of Data Encryption Standard (DES), were proposed [27, 28]. It is reported that the GEs are a little bit more than 2000, and both of them can be used for passive RFIDs. But the possibility to have a collision in three adjacent S-boxes leads to the most successful differential attack based on a 2-round iteration characteristic with a probability of 1/234 [27, 28]. MIBS is a Feistel network cipher and is reported that it can satisfy the requirement for RFID security [29]. Then, despite linear attacks, cipher text attacks and impossible differential attacks do not threaten the full 32-round MIBS but significantly reduces its margin of security by more than 50% [30]. KATAN and KTANTAN have similar properties, and both of them are suitable for resource-constrained devices [31]. KTANTAN48 is the version recommended for RFID tag usage with 588 GEs; the only difference between them is the processing of key scheduling, and slide attacks and related key attacks are also possible to implement; in addition, the related key differential attack is the only attack where there is a difference between the two families of ciphers [32]. In the literature [24], a linear congruential generator (LCG)-based lightweight block cipher was presented, and this cipher can meet security co-existence requirements of WSNs and RFID systems for pervasive computing, but our experiments show that the avalanche effect about it is not good, and thus, it has high possibility to suffer various attacks. TWINE [33] was designed to focus on the requirement of lightweight, and both hardware and software implementations show better performance. To the best of our knowledge, the most powerful attacks are the impossible attacks on 23-round TWINE-80 and 24-round TWINE-128 proposed by the designers and the biclique cryptanalysis of the full cipher [34]. PICO is a substitution- and permutation-based network [35]: the key scheduling is motivated from SPECK cipher, and it does not include a nonlinear layer in the design. PICO shows good performance on both the hardware and software platforms. However, because it is new, there is no further and detailed analysis about the security performance.

In this paper, in the light of the demand of resource-constrained Industrial Wireless Sensor Networks, our target is to study the trade-off between security and performance of several recent top performing lightweight block ciphers. Several software performance metrics are used to evaluate the performance of these ciphers, and the avalanche effect in some ways is adopted to assess security. The contributions of this paper are listed as follows:

- The term "lightweight cipher" is seriously discussed and analyzed considering the implementation platform, and the characteristics of lightweight ciphers are introduced.
- We analyze the Wireless Network for Industrial Automation for Factory Automation (WIA-FA)

security requirement of industrial wireless networks Wireless Network for Industrial Automation (WIA) specification in which the speed and reliability are strict.

- We examine and compare the performance of several carefully selected lightweight block ciphers, and in the meantime, a unified platform is used, and some software specific performance metrics are employed. Despite there are many lightweight block ciphers that are useful and inventive, usually, it is difficult to select a suitable cryptalgorithms for the specific applications. At the same time, the lack of comprehensive and comparative studies brings difficulties and resistances to have a better understanding about the security and performance trade-off.
- The results of avalanche effect, which shows the possibility to resist possible types of different attacks, are presented. When designing a system, the balance between security, cost, and performance has to be accounted [20]. Basically, more iteration rounds and longer key length contribute to a safer system, and the faster and stronger block ciphers would require more costs. However, more rounds mean slowness in algorithms. Overall, we hope that the work of this paper can be served as useful reference for the trade-off between security and performance for further implementation in resource-constrained environments.

The remaining part of this paper is then structured as follows. Section 2 provides a short-related work. Section 3 briefly discusses the characteristics of lightweight block ciphers and some relevant features combined with the security requirements of Industrial Wireless Sensor Networks for Factory Automation. Meanwhile, the implementation details of the better selected lightweight ciphers are also discussed. Section 4 presents the method and dedicated platform, and then, some evaluation metrics are introduced. In Section 4, trade-off between security and performance of these ciphers is analyzed from different aspects, and the avalanche effects, which show the possibilities to resist possible attacks, are also compared. Finally, some conclusions are drawn in Section 6.

2 Related work

There are also some of other papers in the literature that study the trade-off of security and performance [36–40]. The papers [36, 37] study the optimal network performance for stream ciphers. The paper [38, 39] studies the security overhead of aggregation in WIFI. The paper [40] studies the security trade-off of AES over IEEE 802.15.3 wireless personal area networks. In the paper [41], a multilayer authentication protocol and a secure session key generation method are proposed for both security and performance. In the paper [42], a coarse-to-fine clustering method based on a combination of global feature and local feature and PageRank are proposed to nearly eliminate duplicates for visual sensor networks. In the paper [43], optimal cluster-based mechanisms for load balancing with multiple mobile sinks are proposed under the condition of a delay-tolerant application to optimize energy consumption in sensor networks. In the paper [44], an adaptive observation matrix of compressive sensing is proposed for sparse samples for ultrasonic wave signals to reconstruct sensor response signals. In the paper [45], a coverless information hiding method is proposed based on binary numbers to locate the secret information and meet the requirements of both randomness and universality. In the paper [46], relocated mobile sensors to achieve k-barrier coverage with the minimum energy cost is proposed in sensor networks. In the paper [47], a back propagation neural network model using solar radiation to establish its relationship with air temperature error for sensor networks is proposed. In the paper [48], a multilevel pattern mining architecture to support automatic network management by discovering patterns from network monitoring data is proposed.

However, all of the above works are quite different from this paper as explained in the introduction section.

3 Requirements and studied block ciphers

In this section, the characteristics of most of the existing lightweight block ciphers are simply discussed and the differences for requirement of Industrial Wireless Sensor Networks are briefly analyzed. Then, some basic features about WIA-FA are concluded, and the security requirements, which are to meet the strict requirement for speed and reliability in factory automation applications, are discussed. Furthermore, the studied lightweight block ciphers are presented in the following subsection.

3.1 Characteristics of lightweight block ciphers

Generally speaking, the security of such lightweight block ciphers for the resource-constrained environment has their own properties. Usually, security for these applications is just needed to be achieved moderately, and that is to say, the demanded ciphers for constrained environment do not require high-level security, and this is essential in the Internet. On the other hand, attackers in this cryptography environment may be lack of information that is needed to implement cryptanalysis and they themselves can be energy-constrained sometimes, causing the attackers to adopt optimized algorithms and to be smarter enough to effectively implement their attacks [49]. Note that there is no need for lightweight block ciphers to encrypt a great large number of data, and the length of

these data is always delimited into short segments as it is typical in the context of Industrial Wireless Sensor Networks. Lastly, the security performance of each block cipher should be deeply analyzed when the cipher can be practically used in the real environment. Both hardware performance and software performance for lightweight ciphers must be considered, and the hardware performance for some applications, especially for RFID, is the primary consideration. For the specific application, there are some relevant metrics and criteria to measure whether cipher algorithms are good.

In different applications, the security requirement of resource-constrained devices may vary based on the sensitivity of the transmitted data. As for the industry environment, there is a great deal of transmitted data needed to be encrypted in the Industrial Wireless Sensor Networks. Thus, it is essential to require higher throughput for lightweight ciphers, in a sense that sensors in wireless sensor networks usually have more resources such as computation ability, communication ability, and energy compared with RFID tags. Also, because of the software implementations of ciphers do not need additional cost of the hardware manufacturing and often are easy to maintain and upgrade, it is believed that software-oriented implementation of these lightweight ciphers are more practical and useful for sensors.

3.2 Industrial Wireless Sensor Network for Factory Automation (WIA-FA)

Industrial Wireless Sensor Networks, which have characteristics of low cost, easy maintenance, and easy use, are a revolutionary technology. WIA-FA has become the first international wireless technology specification for the applications of high-speed factory automation, and it is a solution by utilizing the 2.4 GHz/5 GHz frequency band to meet the strict requirement for speed and reliability in factory automation applications. Specifically, because of the influence of multifrequency noise, interference, vibration, and multipath effects, it is a problem to realize the reliable communication by utilizing the scarce channel resources. In addition, at the same time, quitting and invalidation of sensor nodes can cause the topology of networks changeable dynamically. From the point of expending, sensor nodes with lower cost usually lead to restrictions on resources of computation and storage. As for the energy consumption, careful measures should be made to guarantee the life span as long as possible. Thus, there is a need for lightweight and lower complexity protocols and algorithms.

The WIA-FA network adopts a centralized management framework, as shown in Fig. 1, and an enhanced star topology is adopted. A host computer is the interface for operators to configure the network and display data. A gateway device is used to achieve interconnection with external networks, and the tasks of network management and security are executed by it. Access devices accept data transmitted from field devices by wireless links, and control commands of gateway device can be forwarded to field devices through access devices at the same time. Field devices can send field application data and alarms to the gateway device, as well as receive configuration information, management information, and control commands from the gateway device.

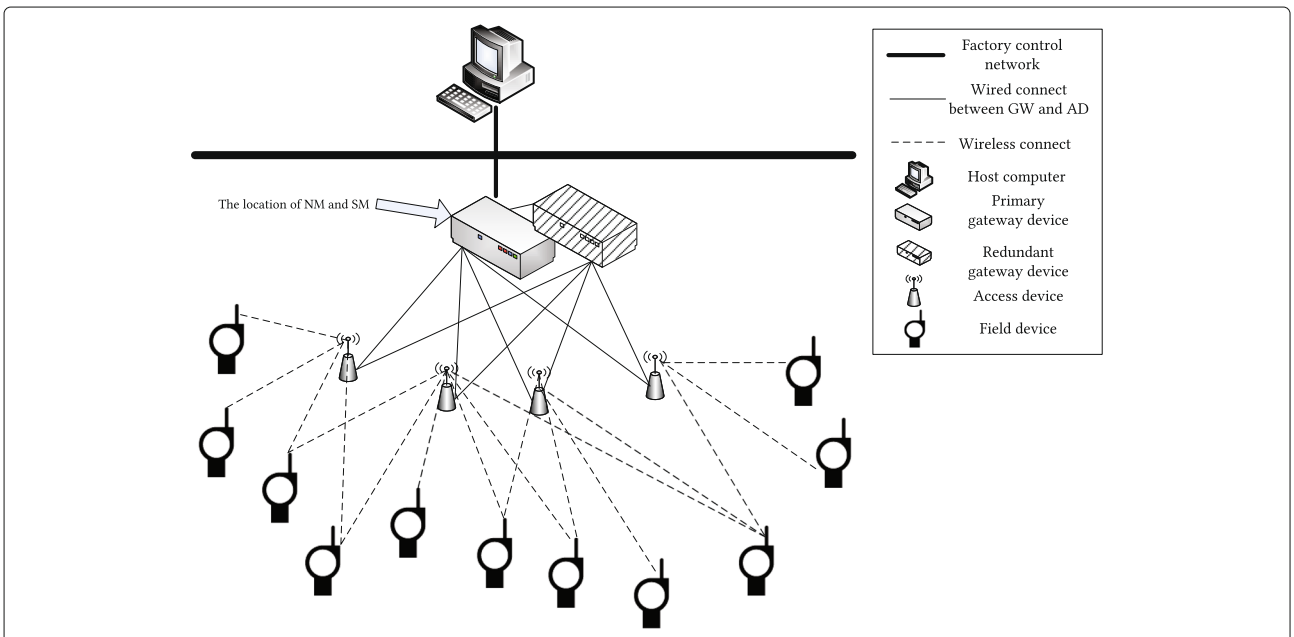

Fig. 1 WIA-FA redundant star topology (legends: NM network manager, SM security manager, GW gateway device, AD access device

As an open system, there are potential inevitable security risks in a WIA-FA network. Therefore, the necessary security measures must be applied to protect the resources within the system and maintain the normal production [50]. Although there is a security framework, encryption algorithms employed are not specified. In addition, according to the characteristics of the WIA-FA network, the security principles, which should be easy to deployment and use, extend battery life, and maximize the use of existing encryption and authentication technologies, are recommended. Based on all these principles and facts of resource-constrained in practice, lightweight block ciphers are imminently needed to fulfil the security requirements. Based on this point of view, the performance and security are implemented and analyzed in this paper.

3.3 The studied block ciphers

The main parameters of block ciphers are block size, key size, and numbers of iteration rounds. Table 1 shows brief collective information about the studied lightweight block ciphers, while some acronyms, background, and implementation details which impact the selection of ciphers for resource-constrained applications of the studied ciphers will be introduced the next.

3.3.1 KLEIN

KLEIN, which is a new family of lightweight block ciphers, is designed for highly resource-constrained devices such as RFID tags and wireless sensor networks. KLEIN was designed as a typical substitution-permutation network (SPN) just as the structure of PRESENT, which is introduced later [51]. In order to obtain a reasonable security level and asymmetric iteration, the number of rounds can be 12/16/20 for the 64/80/96 plaintext, respectively. Since the key length of 64 is a common choice, the KLEIN-64 is adopted in this paper to realize the performance comparison, where the number 64 in the notation KLEIN-64 stands for the key length of 64.

Table 1 List of studied ciphers

Block ciphers	Year	Block size	Key size	Structure	Rounds
KLEIN	2010	64	64/80/96	SPN	12/16/20
LBlock	2011	64	80	Feistel	32
PRESENT	2007	64	80/128	SPN	31
HIGHT	2006	64	128	Feistel	32
Piccolo	2011	64	80/128	Feistel	25/31
SIMON	2012	32/48/64	64/72/96/128		
		/96/128	/144/192/256	-"-	-"-
SPECK	2012	32/48	64/72/96/128		
		/64/96	/144/192/256	-"-	-"-
AES	1998	128	128/192/256	SPN	10/12/14

Algorithm 1: The encryption process of KLEIN

$sk^1 \leftarrow$ KEY;
STATE $\leftarrow$ PLAINTEXT;
for $i = 1$ *to* N^R **do**
  AddRoundKey (STATE, sk^i);
  SubNibbles (STATE);
  RotateNibbles (STATE);
  sk^{i+1} = KeySchedule (sk^i, i);
end
CIPHERTEXT $\leftarrow$ AddRoundKey (STATE, sk^{N^R+1})

The encryption process of KLEIN is shown in Algorithm 1 (Fig. 2) and explained as follows. There are N_R rounds in KLEIN encryption. Each round includes five steps as follows:

- AddRoundKey: The 64-bit plaintext and 64-bit ith round key are XORed with each, where $i = 1, 2, \cdots, N_R$.
- SubNibbles: The XORed result is divided into 16 of 4-bit nibbles, and all of these nibbles are then fed into the same 16 S-boxes. The 4-bit S-box is a 4×4 involution permutation, and it is shown in Table 2. In the meantime, the characteristics of S-boxes satisfy the condition S(S(x)) = x, $x \in F_2^4$, where F_2^4 represents the 4-bit word over the binary field. This property can be used in both the encryption procedure and the decryption procedure. The simple structure with an involution 4-bit S-box playing a role of the nonlinear layer not only assures a better

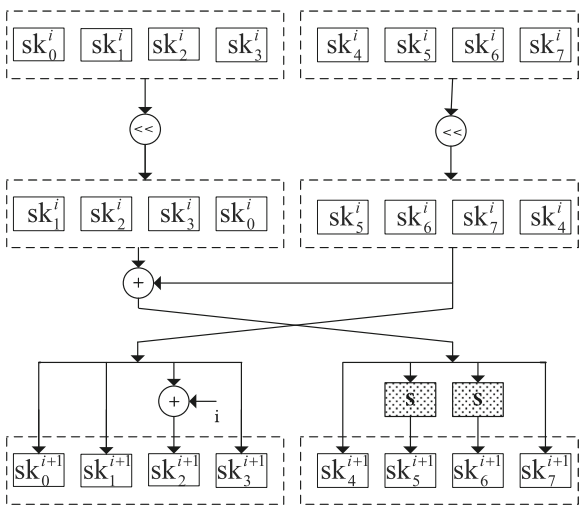

Fig. 2 The key scheduling of 64-bit key length of KLEIN-64

Table 2 The S-box used in KLEIN

Input	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Output	7	4	A	9	1	F	B	0	C	3	2	6	8	E	D	5

implementation of software, but also could resist against linear and differential cryptanalyses.

- RotateNibbles: Two bytes will be circularly rotated left of the 16 output nibbles of the S-boxes. The nibbles will be divided into two tuples. These two tuples are considered as polynomials over F_2^8, where F_2^8 represents the 8-bit word over the binary field. Each of these two polynomials is multiplied by a fixed polynomial $c(x) = 03 \cdot x^3 + 01 \cdot x^2 + 01 \cdot x + 02$. In order to get the results of polynomials of degree less than 4, the above two multiplications are reduced modulo $x^4 + 1$, which is a polynomial of degree 4.
- MixNibbles: The process is similar to the MixColumn step in Rijndael. Because of the unique characteristics of the F_2^8, the addition operation corresponds to the XOR operation between the corresponding bytes in each of the words, and the whole multiplication result can be described as $s'(x) = c(x) \otimes s(x)$, where $s(x) = s_3 \cdot x^3 + s_2 \cdot x^2 + s_1 \cdot x + s_0$ represents the four-term polynomial which is defined with coefficients that are finite field elements, $s'(x) = s'_3 \cdot x^3 + s'_2 \cdot x^2 + s'_1 \cdot x + s'_0$ represents the result of the output in the same form, and the $\otimes$ represents the operation of multiplication. As a result, the four bytes in a tuple are replaced by the following:

$$
\begin{aligned}
s'_0 &= (02 \cdot s_0) \oplus (03 \cdot s_1) \oplus s_2 \oplus s_3 \\
s'_1 &= s_0 \oplus (02 \cdot s_1) \oplus (03 \cdot s_2) \oplus s_3 \\
s'_2 &= s_0 \oplus s_1 \oplus (02 \cdot s_2) \oplus (03 \cdot s_3) \\
s'_3 &= (03 \cdot s_0) \oplus s_1 \oplus s_2 \oplus (02 \cdot s_3)
\end{aligned}
$$

In addition, $s_0 = c^i_{8j+0} \parallel c^i_{8j+1}$, $s_1 = c^i_{8j+2} \parallel c^i_{8j+3}$, $s_2 = c^i_{8j+4} \parallel c^i_{8j+5}$, $s_3 = c^i_{8j+6} \parallel c^i_{8j+7}$, where $j = 0$ or 1 in the different two tuples. $c^i_{8j+k}, k \in [0, 7]$ are the eight four-bit outputs of the ith RotateNibbles step, and the operator $\parallel$ represents concatenation of two 4-bit binary strings. s'_0, s'_1, s'_2, and s'_3 are all the same with eight bits which represent the outputs of the four equations. The output of the MixNibbles step will be the intermediate results for the next round encryption process.

3.3.2 LBlock

LBlock employs a variant of Feistel network, operates on a 64-bit plaintext, supports a key length of 80 bits, and adopts a 32-round iterative structure. The encryption process is illustrated in Fig. 3. A 64-bit plaintext can be described as X0 $\parallel$ X1, where $\parallel$ represents the concatenation of two 32-bit binary strings X0 and X1. For the 32

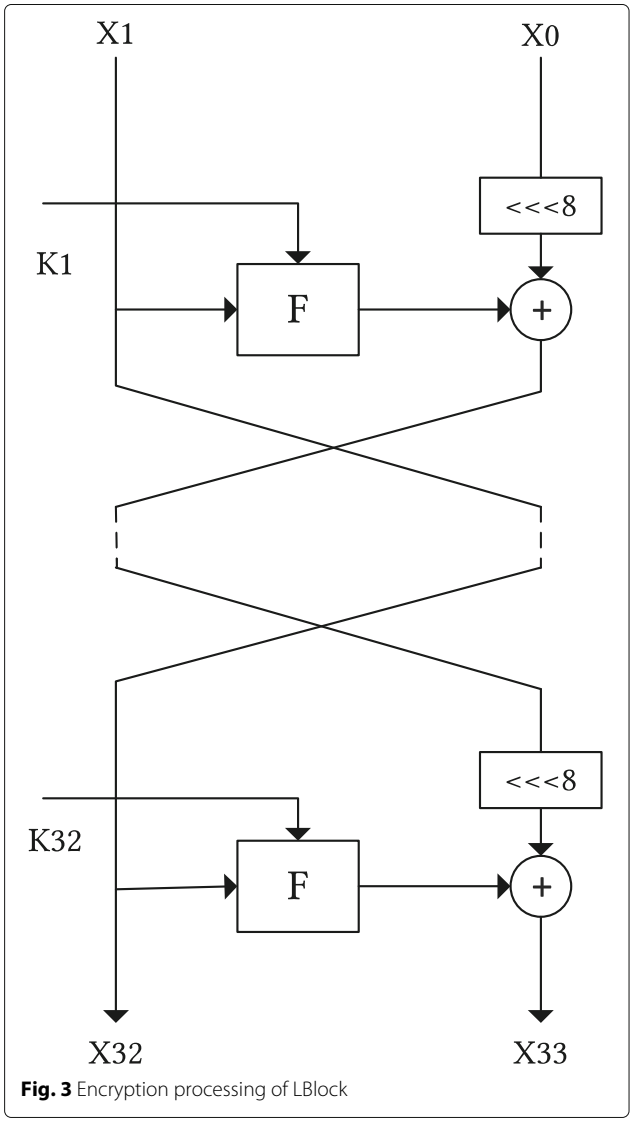

Fig. 3 Encryption processing of LBlock

rounds of data processing, the 32-bit binary strings can be obtained through the equation $X_i = F(X_{i-1}, K_{i-1}) \oplus (X_{i-2} \lll 8)$, $i \in [2, 33]$, where F is the round function which is illustrated in Fig. 4, K_{i-1} represents the 32-bit round subkey in each round encryption, and the operation $\lll$ represents the 8-bit left cyclic shift. In the round function F, in order to achieve the balance between enough security margin and efficient implementation, eight minimized 4-bit S-boxes is shown in Table 3 in which a 4-bit word-wise permutation are used. As shown in Fig. 3, only half of the data are selected to pass through the round function in each round and the other half of the data just use the operation of simply rotation. The key scheduling of LBlock is designed in the way of a stream cipher. Firstly, the round subkey K1 is the output of the leftmost 32 bits of the 80-bit master key $K = k_{79}k_{78}k_{77}k_{76} \ldots k_1k_0$. Then, for $i = 1, 2, \ldots, 31$,

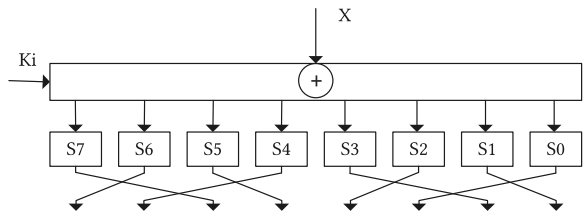

Fig. 4 Round function F

the subkey K_{i+1} is obtained as follows: (1) $k \lll 29$; (2) $[k_{79}k_{78}k_{77}k_{76}] = s_9[k_{79}k_{78}k_{77}k_{76}]$, and $[k_{75}k_{74}k_{73}k_{72}] = s_8[k_{75}k_{74}k_{73}k_{72}]$, where s_8 and s_9 are the two 4-bit S-boxes shown in Table 3; (3) $[k_{50}k_{49}k_{48}k_{47}k_{46}] \oplus [i]_2$, where $[i]_2$ represents the binary form of an integer i; (4) the leftmost 32 bits of the changed K is the round subkey K_{i+1}. Furthermore, the performance evaluation shows that not only hardware is efficient but also software implementation is ultra-lightweight [52]. The original author claimed that LBlock is suitable for RFID tags and sensor networks.

3.3.3 PRESENT

PRESENT is a lightweight cipher which is extremely hardware efficient and was proposed by Bogdanov et al. [53]. Both 80 and 128-bit keys can be used to encrypt a 64-bit plaintext, but usually the version with 80-bit keys is adequate for most low security applications. In many literatures, PRESENT is regarded as a competitive cipher when other lightweight ciphers are designed. PRESENT is a substitution-permutation cipher and 31 rounds iterations are included. The cipher description of PRESENT is showed in Fig. 5. A nonlinear substitution layer, a linear bitwise permutation layer, and a round key K_i where $1 \leq i \leq 31$ are introduced in each of the 31 encryption rounds. Firstly, the 80-bit master key $K = k_{79}k_{78}\ldots k_0$ is stored in the key register, and the 64-bit subkeys K_i in each round are the leftmost 64 bits of the key register.

Table 3 The S-boxes used in LBlock

S-boxes	Value
S0	14, 9, 15, 0, 13, 4, 10, 11, 1, 2, 8, 3, 7, 6, 12, 5
S1	4, 11, 14, 9, 15, 13, 0, 10, 7, 12, 5, 6, 2, 8, 1, 3
S2	1, 14, 7, 12, 15, 13, 0, 6, 11, 5, 9, 3, 2, 4, 8, 10
S3	7, 6, 8, 11, 0, 15, 3, 14, 9, 10, 12, 13, 5, 2, 4, 1
S4	14, 5, 15, 0, 7, 2, 12, 13, 1, 8, 4, 9, 11, 10, 6, 3
S5	2, 13, 11, 12, 15, 14, 0, 9, 7, 10, 6, 3, 1, 8, 4, 5
S6	11, 9, 4, 14, 0, 15, 10, 13, 6, 12, 5, 7, 3, 8, 1, 2
S7	13, 10, 15, 0, 14, 4, 9, 11, 2, 1, 8, 3, 7, 5, 12, 6
S8	8, 7, 14, 5, 15, 13, 0, 6, 11, 12, 9, 10, 2, 4, 1, 3
S9	11, 5, 15, 0, 7, 2, 9, 13, 4, 8, 1, 12, 14, 10, 3, 6

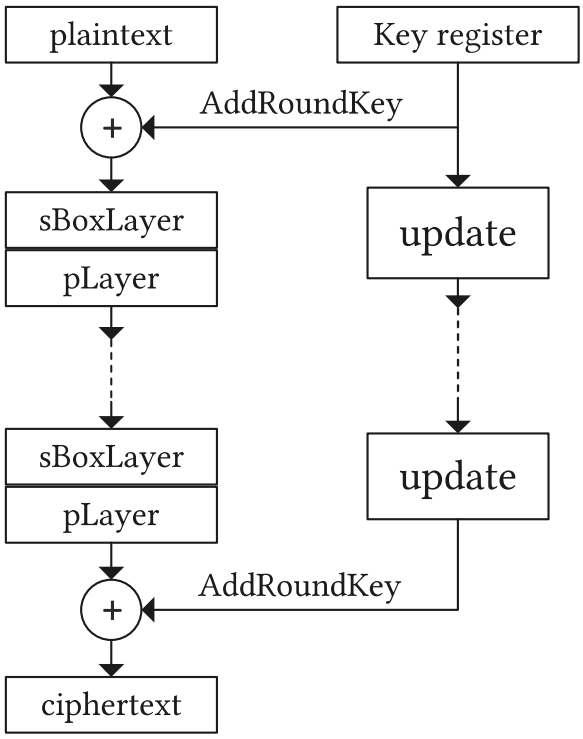

Fig. 5 The cipher description of PRESENT

Then, the contents of the key register is updated as follows: (1) the key register is rotated by 61 bits to the left, i.e., $[k_{79}k_{78}\ldots k_1, k_0] = [k_{18}k_{17}\ldots k_{20}k_{19}]$; (2) the leftmost four bits are substituted by the S-box shown in Table 4, i.e., $[k_{79}k_{78}k_{77}k_{76}] = S[k_{79}k_{78}k_{77}k_{76}]$; (3) the round counter i is XOR with bits $k_{19}k_{18}k_{17}k_{16}k_{15}$ to replace the original value, where i uses the binary form. After the 64-bit round keys are obtained, the intermediate states in each round are XOR with round keys in the step of AddRoundKey, and then, the 64-bit current states are considered as sixteen 4-bit words which are replaced by the mentioned S-box. Finally, the 64-bit states are permutated by a specific permutation table (i.e., pLayer in Fig. 5), and the subkey K_{32} is used for post-whitening through the step of AddRoundKey. It is reported that implementation results can realized as low as 1570 gate equivalent at the hardware level. But the software performance is the point that we mainly care about just because of the sensor nodes have abundant resources than RFID tags and the software implementation is easy to update and modify on different platforms.

Table 4 The S-box used in PRESENT

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

3.3.4 HIGHT

The block cipher of HIGHT has a 128-bit master key with a 64-bit block length based on a variant of generalized Feistel network. HIGHT was designed by Hong et al., and a 32-round iteration is implemented during the encrypt process [54]. The 128-bit master key $MK = MK_{15} \parallel \ldots MK_0$ is a concatenation of 16 bytes, where MK_i, $i \in [0, 15]$ represents the byte. The whole encryption process of HIGHT is shown in Fig. 6 with the steps of key schedule, initial transformation, round function, and final transformation. During the process of encryption, only the 128-bit master key is required to store, and eight whitening keys WK_i, where $i \in [0, 7]$ and the sub-keys SK_i, where $i \in [0, 127]$, can be generated on the fly. Eight whitening keys in total are used for initial and final transformations, and four sub-keys are used for the computation in each round. In the step of initial transform, the 8-bit plaintext $P = P_7 \parallel \ldots P_1 \parallel P_0$ is transformed by using the four whitening keys into the 8-bit input $X_0 = X_{0,7} \parallel \ldots X_{0,1} \parallel X_{0,0}$ of the first round as follows: $X_{0,0} \leftarrow P_0 \boxplus WK_0$, $X_{0,1} \leftarrow P_1$; $X_{0,2} \leftarrow P_2 \oplus WK_1$, $X_{0,3} \leftarrow P_3$; $X_{0,4} \leftarrow P_4 \boxplus WK_2$, $X_{0,5} \leftarrow P_5$; $X_{0,6} \leftarrow P_6 \oplus WK_3$, and $X_{0,7} \leftarrow P_7$, where the operation $\boxplus$ represents the addition mod 2^8 and operation $\oplus$ represents the exclusive-or (XOR), respectively. In the step of 32 round functions, the intermediate results $X_i = X_{i,7} \parallel \ldots X_{i,1} \parallel X_{i,0}$ will be transformed into $X_{i+1} = X_{i+1,7} \parallel \ldots X_{i+1,1} \parallel X_{i+1,0}$ where $i = 0, 1, \ldots, 31$ as follows: $X_{i+1,1} \leftarrow X_{i,0}$, $X_{i+1,3} \leftarrow X_{i,2}$, $X_{i+1,5} \leftarrow X_{i,4}$, $X_{i+1,7} \leftarrow X_{i,6}$, $X_{i+1,0} = X_{i,7} \oplus (F_0(X_{i,6})) \boxplus SK_{4i+3}$, $X_{i+1,2} = X_{i,1} \boxplus (F_1(X_{i,0})) \oplus SK_{4i+2}$, $X_{i+1,4} = X_{i,3} \oplus (F_0(X_{i,2})) \boxplus SK_{4i+1}$, $X_{i+1,6} = X_{i,5} \boxplus (F_1(X_{i,40})) \oplus SK_{4i}$, where the functions $F_0(x) = (x \lll 1) \oplus (x \lll 2) \oplus (x \lll 7)$, $F_1(x) = (x \lll 3) \oplus (x \lll 4) \oplus (x \lll 6)$, and the operation $\lll$ represents the bit left rotation of a 8-bit value. Finally, in the step of final transform, the ciphertext $C = C_7 \parallel \ldots C_1 \parallel C_0$ can be obtained by the last round function result $X_{32} = X_{32,7} \parallel \ldots X_{32,1} \parallel X_{32,0}$ as follows: $C_0 \leftarrow X_{32,1} \boxplus WK_4$, $C_1 \leftarrow X_{32,2}$, $C_2 \leftarrow X_{32,3} \oplus WK_5$, $C_3 \leftarrow X_{32,4}$, $C_4 \leftarrow X_{32,5} \boxplus WK_6$, $C_5 \leftarrow X_{32,6}$, $C_6 \leftarrow X_{32,7} \oplus WK_7$, and $C_7 \leftarrow X_{32,0}$. Because some simple operations such as XOR and bit-wise rotation are adopted, this cipher is efficient to be hardware-oriented. Furthermore, the designer of HIGHT claimed that the software implementation of HIGHT is faster compared with AES-128. Differential cryptanalysis, linear cryptanalysis, saturation, and boomerang attack analysis show better performance about HIGHT. Moreover, the strength of its security is described to be abundant on account of the NIST statistical test result.

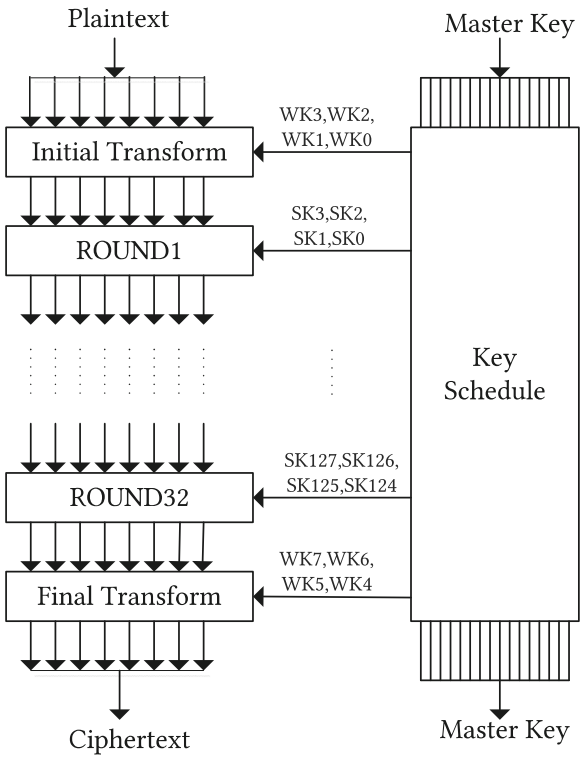

Fig. 6 The encryption process of HIGHT

3.3.5 Piccolo

Piccolo, which is an lightweight block cipher, shows both high security and compact hardware implementation. Piccolo supports 64-bit block to fit standard applications, and 80 or 128-bit keys to achieve moderate security levels [55]. The structure of Piccolo is a variant of generalized Feistel network, and the encryption function is described as Fig. 7. Here, we only focus on the Piccolo 64–128, which consists of 31 rounds with 64-bit plaintext and a 128-bit master key. The encryption process is described as follows. Firstly, the 64-bit plaintext $X = X_0 \parallel X_1 \parallel X_2 \parallel X_3$ combined with four 16-bit whitening keys wk_i, $i \in [0, 3]$ and sixty-two 16-bit round keys rk_i, $i \in [0, 61]$ are the inputs of the encryption, where the bit length of X_i, $i \in [0, 3]$ are all the 16 bits. For the start of the encryption, $X_0 \leftarrow X_0 \oplus wk_0$, and $X_2 \leftarrow X_2 \oplus wk_1$, where the notation $\leftarrow$ means updating a value, and $\oplus$ means the operation of XOR. Then, for each round $i \in [0, 29]$, the round function is implemented as follows: $X_1 \leftarrow X_1 \oplus F(X_0) \oplus rk_{2i}$, $X_3 \leftarrow X_3 \oplus F(X_2) \oplus rk_{2i+1}$, $X_0 \parallel X_1 \parallel X_2 \parallel X_3 \leftarrow RP(X_0 \parallel X_1 \parallel X_2 \parallel X_3)$, where the function F is showed in Fig. 8, and the function RP represents the round permutation operation $(x_0, x_1, \ldots, x_7) \leftarrow (x_2, x_7, x_4, x_1, x_6, x_3, x_0, x_5)$ in which each of the x_i, $i \in [0, 7]$ is eight bits to divide the 64-bit intermediate value. Finally, the whitening keys wk_2 and wk_3 are used for the operations of $X_0 \leftarrow X_0 \oplus wk_2$,

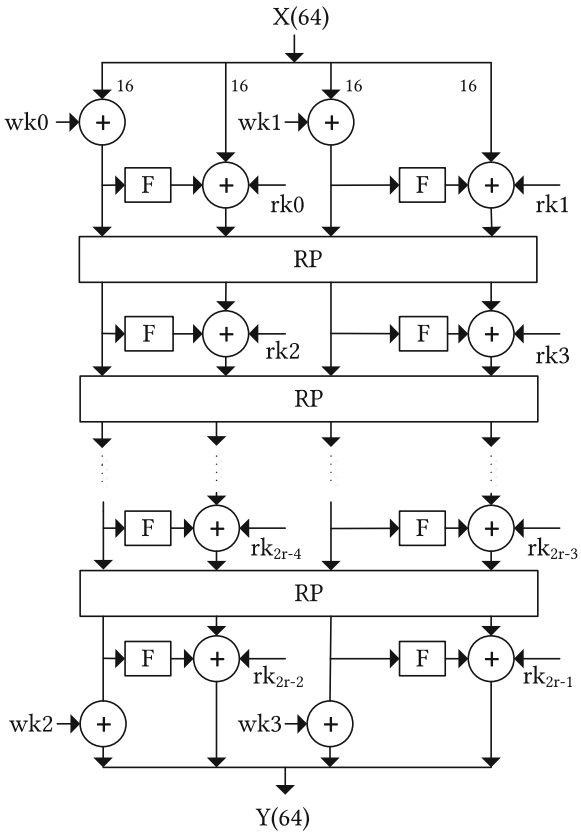

Fig. 7 The encryption process of Piccolo

and $X_2 \leftarrow X_2 \oplus wk_3$. The function F consists of two S-box layers and a diffusion matrix M in which the 4-bit S-box is presented in Table 5. The diffusion matrix M is defined as $M = \begin{bmatrix} 2 & 3 & 1 & 1 \\ 1 & 2 & 3 & 1 \\ 1 & 1 & 2 & 3 \\ 3 & 1 & 1 & 2 \end{bmatrix}$

The computation for the 16-bit data is defined as $(x_0, x_1, x_2, x_3)^T \leftarrow M \cdot (x_0, x_1, x_2, x_3)^T$, where the notation T represents the transposition of a vector, x_i, $i \in [0, 3]$ are 4-bit data which are obtained by the outputs

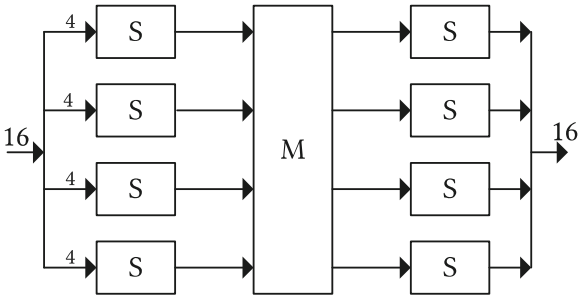

Fig. 8 *F* function of Piccolo

Table 5 The S-box used in Piccolo

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

of the S-boxes, and the multiplication is performed over Galois field $GF(2^4)$ defined by an irreducible polynomial $x^4 + x + 1$. The authors claimed that the hardware implementation requirements for the 80-bit and 128-bit key modes were 683GE and 758GE, respectively, when this cipher was compared to general standers of 2000 gate equivalents.

3.3.6 SIMON and SPECK

The SIMON and SPECK families of block ciphers were proposed publicly by the NASA in 2013 [56]. The motivation of the design is the security requirement of sufficient flexibility in the new area of Internet of Things because all the tiny devices in heterogeneous networks will require adequate cryptography, and the most existing block ciphers with fixed block size are lack of flexibility on different application platforms. Both SIMON and SPECK have multiple instantiations, supporting block sizes of 32, 48, 64, 96, and 128 bits, and with up to three key sizes to go along with each block size. The author claimed that SPECK has the highest throughput in software compared with any block ciphers in the literature and SIMON have the best performance in hardware performance. Thus, to our purpose, we only focus on the features of the SPECK. In the design aspect, the SPECK round functions are based on the Feistel structures and S-boxes are not used so that a good balanced between linear diffusion and nonlinear confusion can be achieved.

In Fig. 9, the round function of SPECK is the map as follows: $(X_{2i+3}, X_{2i+2}) \leftarrow \left(\left(S^{-\alpha}X_{2i+1} + X_{2i}\right) \oplus k_i, S^{\beta}X_{2i} \oplus \left(\left(S^{-\alpha}X_{2i+1} + X_{2i}\right) \oplus k_i\right)\right)$, where X_{2i} and X_{2i+1} are the inputs of n-bit quantities, k_i represents the ith round key, the parameters α and β are 8 and 3, respectively, except for the case of SPECK 32/64 in which the parameters α and β are 7 and 2, and the operations S^j represents the left circular shift by$j$ bits. As for the key scheduling of SPECK, the round function is reused and this promotes the reduction in the amount of code size, which is what the resource-constrained devices prefer. SPECK's key schedule are presented as follows: $l_{i+m-1} = (k_i + S^{-\alpha}) \oplus i$; $k_{i+1} = S^{\beta}k_i \oplus l_{i+m-1}$, where the parameter i is the round counter, the parameters α and β which represent the number of left circular shifted bits, are 8 and 3, respectively, and m is the number of words of key and at the same time, the key can be written as $(l_{m-2}, \ldots, l_0, k_0)$. All of these characteristics are also what we need in our subsequent application, and SPECK64-128 and SPECK128-128 are what we focus on because the length of the plaintext and keys are usually used when compared with other lightweight ciphers.

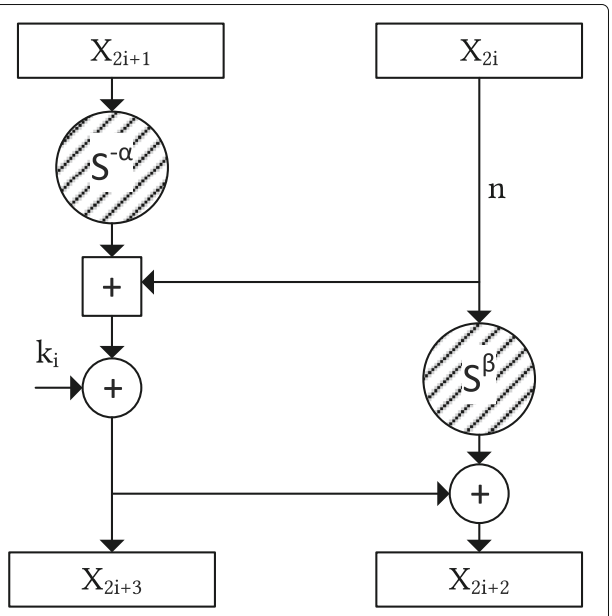

Fig. 9 Round function of SPECK

3.3.7 Advanced encryption standard

Advanced Encryption Standard (AES) has a great deal of impact in the modern cryptography, and it is widely used in many applications because of many better characteristics when compared with other ciphers such as stream ciphers and asymmetric cryptography. It has been created to achieve good performance both in hardware and in software. AES is based on a substitution-permutation network structure and the block length is 128 bits with the length of 128, 192, or 256 bits keys [40, 57–59]. Normally, the AES of 128-bit keys is a sufficient selection for different usage with different purpose. In each round of the multiple iterations, the operations of SubBytes, ShiftRows, MixColumns, and AddRoundKey are included. For resource-constrained devices, AES could be too expensive to use despite there are various approaches that had been proposed to reduce the costs during the implementation of hardware and software. Here, we use the widely used cipher AES to achieve the purpose of comparison of these lightweight block ciphers.

4 Methods/experimental

A lightweight cipher in some ways is defined as a crypt algorithm that is specifically designed for resource-constrained devices, and three challenges minimal overhead, lower power consumption, and adequate security level must be balanced. However, to a certain extent, the term lightweight is usually overused because of a great deal of different definitions in different literature. A good way to solve this problem and a more objective method to compare the performance of existing different lightweight block ciphers is to use a uniform platform. From this point of view, these ciphers were implemented on a specific platform, and the information about it is present as follows. Then, metrics to measure performance of these lightweight block ciphers are discussed, and five specific indicators are listed in Table 6. In the end of this subsection, different compiling modes are slightly analyzed because it is important in the continued work.

4.1 The dedicated platform

A STM32F407ZGT6 is used, and it has a 32-bit reduced instruction set computer (RISC) micro-controller at a frequency of up to 168 MHz with the high performance ARM Cortex-M4 core. A floating-point unit can support all ARM single precision data processing instruction data types. As for the memories, the flash memory can be up to 1 Mbyte, and the static random-access memory (SRAM) can reach 192 Kbytes. All devices offer three 12-bit analog digital converters (ADCs), two digital to analog converters (DACs), a low-power real-time clock (RTC), and 12 general purpose 16-bit timers. Standard and advanced communication interfaces such as Inter-Integrated Circuit bus (I2C), Serial Peripheral Interface (SPI), Inter-IC Sound (I2S), and Universal Asynchronous Receiver Transmitter (UART) are included. Moreover, there are rich I/O (input or output) interfaces to provide many peripheral functions. The power supply can be 1.8 to 3.6 V, and a comprehensive set of power-saving mode allows the usage of low power consumption. All of these features make the controller suitable for a wide range of applications, especially for the purpose of industry environment.

To implement these ciphers on such a platform, all the codes were written in C through the new vision Real View MDK5.14, and the uVision5 integrated development environment was used. As for the debugger, J-LINK was used to flash programs into the micro-controller, and the options for different compiling modes were selected to test the performance of these lightweight block ciphers.

4.1.1 Software platform metrics

It is well known that performance metrics play an important role when different cipher algorithms are compared. Hence, it is not accurate in the same study to compare

Table 6 Software implementation performance metrics

Metric	Definition
Code size (bytes)	Memory size to store the cipher code and constants. Typically, resides in flash memory.
RAM size (bytes)	Memory size to store the intermediate states during the execution of the cipher code.
Cycles/byte	Number of cycles to encrypt (decrypt or both) one block.
Throughput	Number of encrypted bits per second (Kbps).
Combined metric	Code size × *Cycle*_count/Block_size.

ciphers implemented in different environment, and further inaccuracies can be introduced when a metric is estimated from other metrics. Consequently, a uniform platform and consistently agreed on metrics are needed.

To our knowledge, the metrics for software and hardware implementations are not identical because the implementation complexities of the cipher operations are different in software and hardware [60]. The implementation of bit permutation is expensive in software but it is easy to implement in hardware, and in practice, large look-up tables can be very easy to set up in software but it may become extremely tough in hardware. The basic performance metrics for hardware designs are area, timing, and energy. Additionally, there are composite metrics in hardware, such as the power and the efficiency metrics [61]. However, as is mentioned in some recent studies, software implementations have more mature performance metrics and measurements. Usually, a microprocessor is only needed to operate software implementations. The main design goals are to reduce the memory occupation and to optimize the throughput and power saving. In addition, obviously, portability is a main advantage compared with hardware implementations. Here, we only focus on the software platform metrics.

Some specific metrics that we use are listed as follows [61]. Typically, the complexity of an algorithm is usually combined space complexity and time complexity. Based on this start point, code size and random-access memory (RAM) size are used to describe the occupancy of the micro-controller's space. Cycles/byte is defined as number of processor's cycles to deal with one block, and throughput is defined as a function combined process's frequency with cycles/byte. Both of them can be seen as the metrics of the complexity of time. As for the combined metric, in a sense it is a more fair mechanism because of the code size and the time consumed are both considered. However, when the lightweight ciphers are specifically implemented, metrics of the performance evaluation should be chosen according to the actual situations.

While most of the researchers only focus on the process of the encryption because of the operations about encryption and decryption are constantly similar, especially for the involution ciphers, we consider the implement of these lightweight ciphers both encryption and decryption architecture, and the algorithms of key scheduling just in the purpose of different kinds of operations could be included.

4.1.2 Different compiling modes

During the software implementations of different lightweight block ciphers, different compiling modes could cause a big diversity. In total, the ARM Compilation Tools offer a range of options to apply when compiling cipher codes, the term -O3 and -Os represent the optimization of the focus on achieving the best performance of time and the smallest code size, respectively. Cross-module optimization has been used, and it shows to reduce code size by removing unused functions from the application. It can also improve the performance by allowing modules to share inline code. The combination of options applied will depend on the optimization goal to meet specific requirements.

5 Results and discussions

In this section, using the previously defined methodology, the software implement results in different compiling modes of these lightweight block ciphers are presented. These ciphers are all proposed recently, and we evaluate performance of them in different aspects, which can help to make good decisions in the situations of complex industrial applications and resource-constrained environments. In addition to the memory requirements, the minimized execution time of lightweight block ciphers is the point that were most concerned about, and the information about this to some extent could be found from the metrics of throughput and cycles/byte. Finally, a relatively comprehensive result regarding algorithm efficiency combined code size and complexity of execution speed is described. In addition, during the discussion of each subsection, comparisons and analysis are presented in detail.

5.1 Memory occupation

As compact implementation is one of the primary goals for resource-constrained devices, the memory sizes are compared under different modes, and among which the optimization of the smallest code size is preferred. What is mentioned from [62] is just precisely described as follows: ultra-lightweight implementations require up to 4 KB ROM and 256 bytes RAM, low-cost implementations require up to 4 KB ROM and 8KB RAM, and lightweight implementations require up to 32 KB ROM and 8 KB RAM. These targets make sure that ciphers can be used in a variety of platforms. RAM is used to store the stack and variables of intermediate calculation results, and some zero-initialized variables are also stored in RAM on the STM32F407 platforms. Because of the characteristics and distinctive architecture of RAM in the dedicated micro-controller, the source programs are first downloaded to the flash memory which could speed up the code execution. Only the stacks and zero-initialized variables of the system information are stored in the RAM, and thus, the RAM occupations of all these lightweight ciphers in both of the following two modes are the same value.

As illustrated in Figs. 10 and 11, both of the ciphers SPECK64_128 and SPECK128_128 have the smallest flash memory using less than 1700 bytes; the memories need for HIGHT and Poccolo64_128 are almost equivalent;

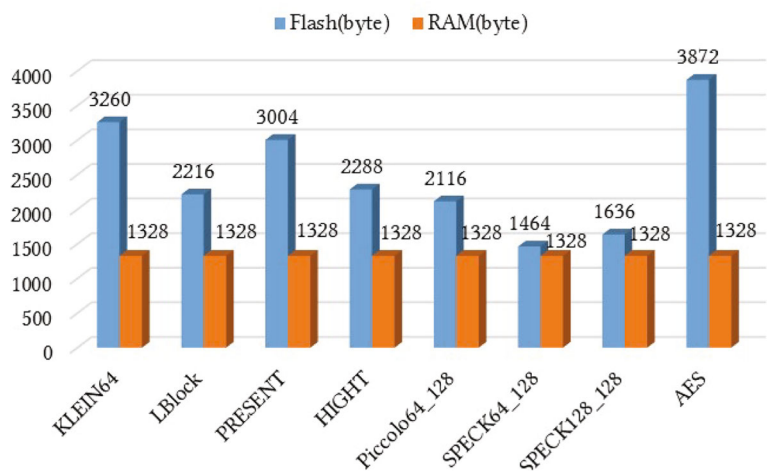

Fig. 10 Memory usage of mode -O3

LBlock, KLEIN-64, and PRESENT are relatively poor than the abovementioned ciphers. Usually, PRESENT is used as a benchmark for newer lightweight ciphers, the gate equivalents is less to 1570 GEs, and thus, it is hardware-oriented. However, the software footprint about PRESENT is conversely slightly higher. The reason that KLEIN-64 has high memory is from the facts that elementary operations were borrowed from AES and PRESENT. Obviously, the standard cipher AES occupies the most resources even though 2932 bytes of flash memory are needed in the mode of -Os though a matrix of bytes is used to represent tables for operations of ShiftRows and MixColumns. The above presentations are surely what the target of low-cost devices is expected, and the less space occupied, the wider scope the applications have.

5.1.1 Throughput and cycles/byte

Since there is no such a direct instruction in the selected platform which can be adopted to measure the processing speed by throughput and cycles per byte, the speed is compared by using the results of processing a block of plaintext combined with the key scheduling and the obtained numerical values are calculated and listed as follows. The throughput is usually expressed to describe the number of processed bits per second, depending on the processor's frequency and the instruction set. In time-critical applications, delay could cause serious consequences. Especially in the industrial environments, the speed of data processing and data transmitting is an important index, which may cause delay and errors in the production process. Notably, in the case

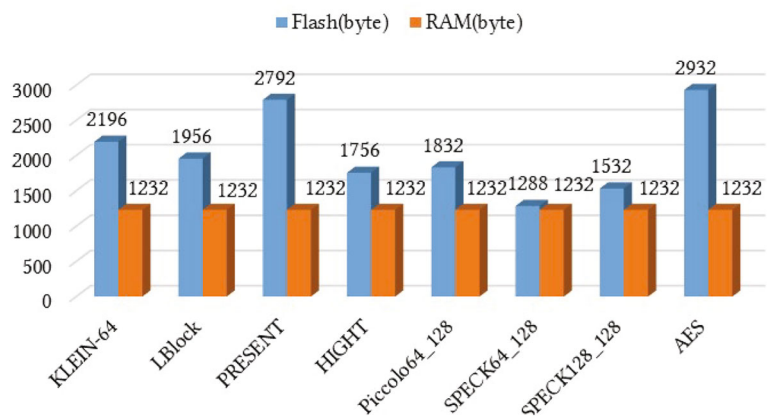

Fig. 11 Memory usage of mode -Os

of time optimization from Fig. 12, the throughputs of SPECK64_128 and SPECK128_128 are extremely large, and they are consistently the top performers as their designers mentioned. AES follows and offers a good speed, just like being verified by a lot of standard platforms. KLEIN-64, LBlock, and HIGHT are slower than AES. Because PRESENT is a hardware-oriented lightweight block cipher with the tiny 4-bit S-box and all the designs enable the minimal and compact hardware implementation, the speed of software throughput is relatively low. Furthermore, the space optimization mode from Fig. 13 shows the similar results.

On the contrary, cycles/byte, which express the cycles needed to deal with one byte, is an opposite metric to measure the performance of processing speed. Figures 14 and 15 show that results conform with the above observations.

Summaries of the above results are presented in Tables 7 and 8.

5.1.2 Comprehensive metric

The combined metric is defined in Table 6 as Code-size×*Cycle*_count/Block_size. A smaller value of comprehensive metric indicates a better lightweight cipher [22]. Figures 16 and 17 show trade-off of code size versus performance in terms of speed. Among these eight ciphers, the ranking order of ciphers from good to bad is SPECK64_128, SPECK128_128, HIGHT, LBlock, KLEIN64, AES, Piccolo64_128, and PRESENT. SPECK64_128 is the best and is followed by SPECK128_128. AES exhibits a little bit bad characteristics for both the optimization modes of space and time. HIGHT and LBlock which are smaller than AES present a slightly good trade-off between code size and cycles count. PRESENT is relatively large than AES as shown in figures, because it is hardware-oriented. The Piccolo64_128 is also worse than that AES. In summary, the ciphers SPECK64_128 and SPECK128_128 achieve the best comprehensive metrics and are the best choices for resource-constrained devices such as wireless sensor networks, especially for the real-time applications. We also observe that there are sufficient space for a trade-off between security and cost.

5.1.3 Avalanche effect comparison

Avalanche effect is an important characteristic for block ciphers. It is defined as the fact that with change in a single bit of a plaintext or a key, many bits will change in the corresponding ciphertext. Initially, the avalanche effect is used to measure the amount of nonlinearity in the substitution box, which is a crucial component of many block ciphers. Subsequently, it also can be employed to measure the processing functions of the encryption. The avalanche effect tries to reflect, to some extent, the intuitive idea of high nonlinearity. Mathematically,

$$\forall x, y \in Z_2^n \mid H(x,y) = 1, \text{average } H(F(x), F(y)) \geq \frac{n}{2}$$

where x and y are two vectors for the input of the encryption, and H represents the Hamming distance function, which is defined as the number of positions where the vectors differ. Usually, it can be defined as the number of ones of vector $z = x \oplus y$. Therefore, this formula shows that if F has a better avalanche effect, the Hamming distance between the outputs of a random input vector and one generated by randomly flipping one of its bits should be at least $\frac{n}{2}$ on average [63].

Ciphers which possess good avalanche effect have higher possibility to resist various different attacks, and thus an attacker is quite difficult to conduct analysis of cipher text when attacks are launched. The results obtained from Fig. 18 reflect the avalanche effects of these

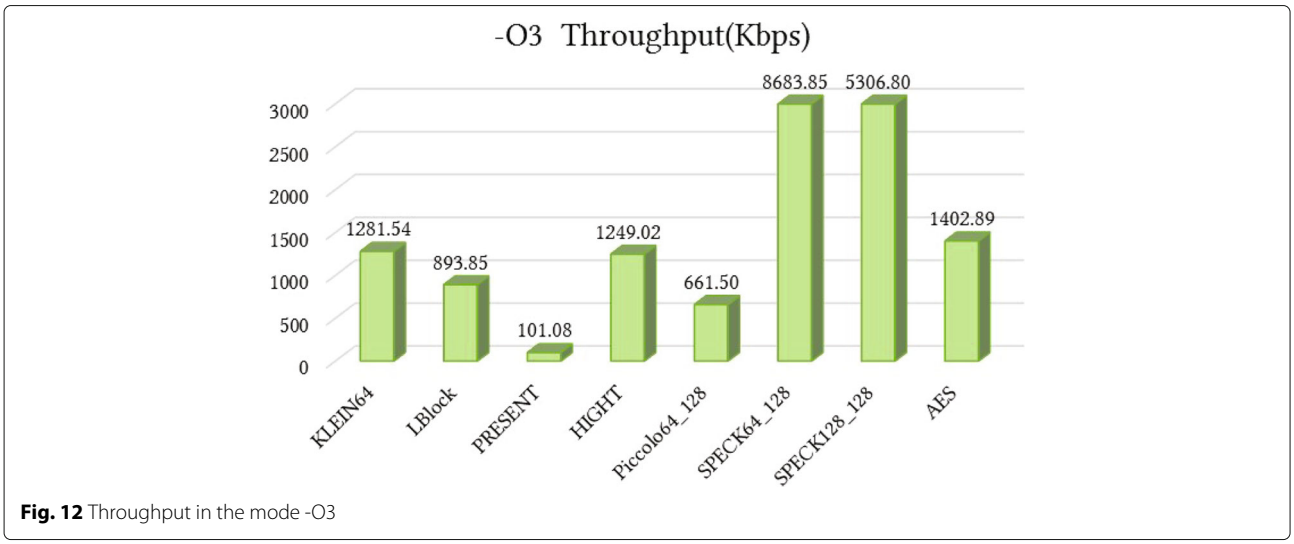

Fig. 12 Throughput in the mode -O3

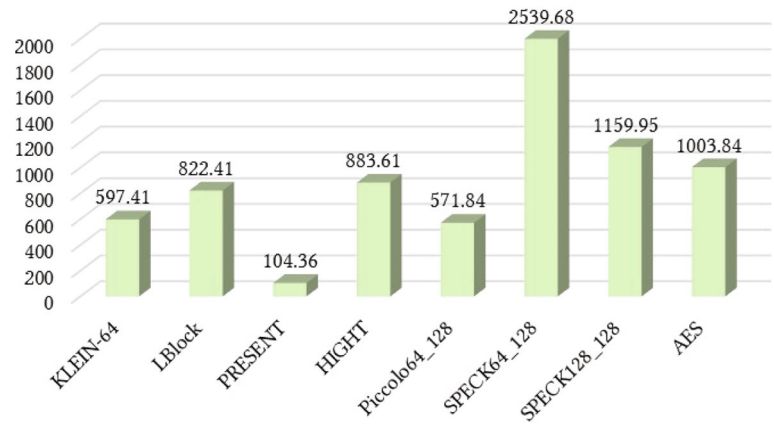

Fig. 13 Throughput in the mode -Os

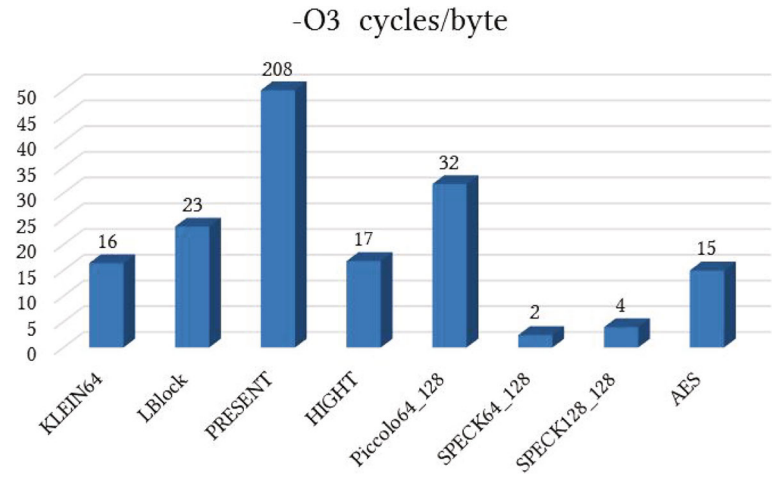

Fig. 14 Cycles/byte in the mode -O3

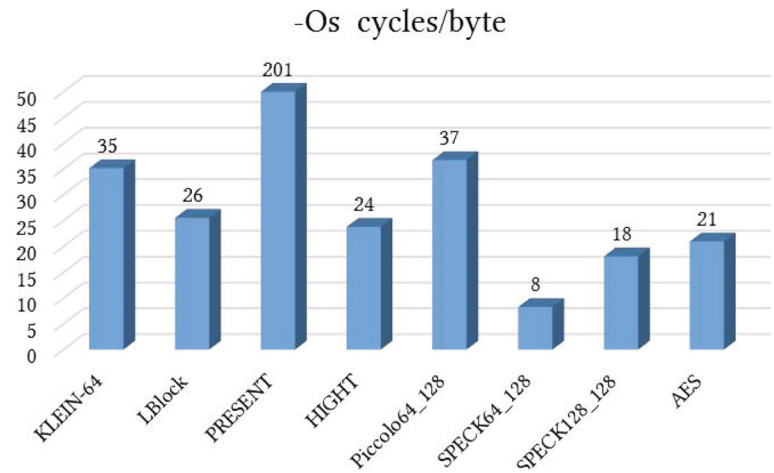

Fig. 15 Cycles/byte in the mode -Os

Table 7 Performance of mode -O3

Block ciphers	Flash (byte)	RAM (byte)	Exec. time (μs)	Throughput (Kbps)	Cycles/byte
KLEIN-64	3260	1328	49.94	1281.54	16
LBlock	2216	1328	71.60	893.85	23
PRESENT	3004	1328	633.18	101.08	208
HIGHT	2288	1328	51.24	1249.02	17
Piccolo64_128	2116	1328	96.75	661.50	32
SPECK64_128	1464	1328	7.37	8683.85	2
SPECK128_128	1636	1328	24.12	5306.80	4
AES	3872	1328	91.24	1402.89	15

lightweight block ciphers. It is observed that PRESENT is relatively worse than the other algorithms. Related key attacks and slide attacks are the most effective attacks to PRESENT [51, 53], and although the hardware implementation is competitive, the software performance combined memory, throughput, and the comprehensive metric shows that PRESENT is not appropriate when it is implemented in resource-limited devices since the security and software performance both are not good. Results from the above show that LBlock is lightweight in term of the memory occupation, and the throughput in mode -O3, is better than PRESENT. But differential cryptanalysis is one of the possible attacks of LBlock [52]. As for KLEIN-64, the operations RotateNibbles and MixNibbles help to achieve a balance between the minimum number of active S-boxes and the software performance. However, there is also an integral attack that can be mounted based on the 15-round integral distinguisher [51]. The needed memory of HIGHT is smaller than AES, and the comprehensive metric of HIGHT is better than AES. The Boomerang attack is an applicable on 11-round of HIGHT. SPECK64_128 and SPECK128_128 are the two ciphers, which show excellent performance in various aspects, extremely smaller memory, higher throughput, faster speed than other ciphers, and have pretty good comprehensive metrics. The avalanche effect about them is good, and to date, all published attacks on SPECK are of the reduced-round variety. One of the measures of security about block ciphers is the number of rounds that can be attacked among the total rounds. For SPECK, there is no published attack that can make this percentage more than 70% of all rounds for all versions of SPECK [64]. In other words, SPECK has a relatively satisfactory security performance. However, despite all that, all of the lightweight block ciphers can be used in the situations which security is not much concerned. In practice, one of the industry requirements is that tasks are performed in a timely manner. Actually, block ciphers which have long keys or large rounds enhance the security and correspondingly decrease the real-time performance on the contrary. Thus, the real-time performance and the high security requirement contradict each other. Lightweight ciphers should be carefully selected for the specific purpose with the considerations that the specific platform which may be resource constrained, and this is the focus of attention in the industry wireless environment.

6 Conclusions

Based on the fact that both security requirements and performance of lightweight block ciphers should take into careful consideration in Industrial Wireless Sensor Networks, this paper studied several recent top performing lightweight ciphers on a specific low-cost platform. Some software-oriented performance metrics are used to

Table 8 Performance of mode -Os

Block ciphers	Flash (byte)	RAM (byte)	Exec. time (μs)	Throughput (Kbps)	Cycles/byte
KLEIN-64	2196	1232	107.13	597.40	35
LBlock	1956	1232	77.82	822.41	26
PRESENT	2792	1232	613.27	104.36	201
HIGHT	1756	1232	72.43	833.61	24
Piccolo64_128	1832	1232	111.92	571.84	37
SPECK64_128	1288	1232	25.20	2539.68	8
SPECK128_128	1532	1232	110.35	1159.94	18
AES	2932	1232	127.51	1003.84	21

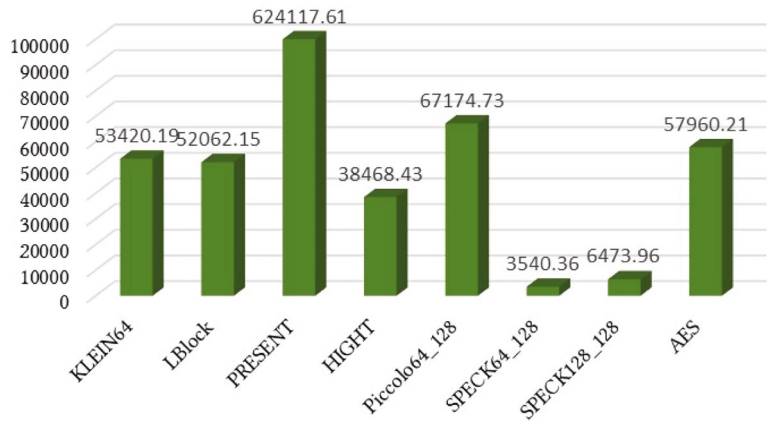

Fig. 16 Comprehensive metric of mode -O3

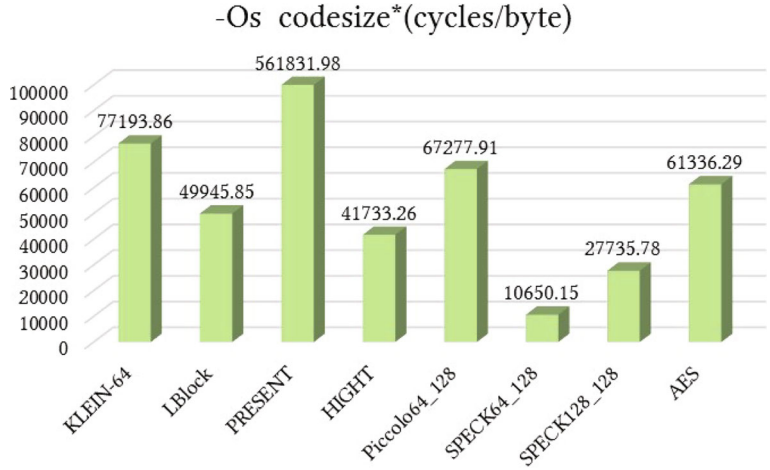

Fig. 17 Comprehensive metric of mode -Os

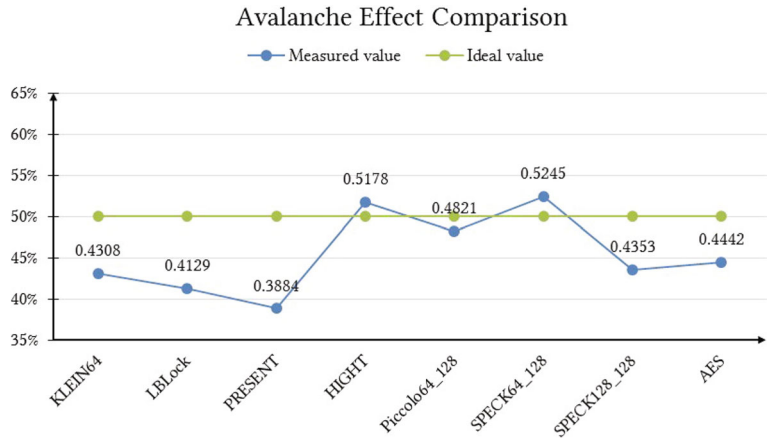

Fig. 18 Avalanche effect for different block ciphers

measure the performance of these ciphers from different aspects. In addition, the avalanche effect defined as the possibility to resist possible types of different attacks indicates the security characteristics of these ciphers. Through the analysis and comparison of experimental data results, it is obvious that the cipher SPECK shows good competitiveness in various aspects, such as the least memory occupation, the highest throughput, the best comprehensive metric, and a better security.

In addition, although PRESENT with the nature of compact hardware implementation is usually served as a benchmark for newer hardware-oriented lightweight ciphers, the software performance combined with avalanche effect is inadequate when it is implemented. Thus, the balance between security and performance has to be paid attention to when a system is designed to achieve the expected results.

Usually, in actual applications, the environment of Industrial Wireless Sensor Networks is extremely complex with strict requirements such as speed and reliability that are strict as the precondition to guarantee the stable operations of the system. Thus, to select a suitable cryptographic algorithm optimized to the factory environment, there is a need to better understand the resources of dedicated platforms and the algorithmic requirement. Nice trade-off between security and performance will help to put forward good solutions to actual applications. Scopes for further research include the lightweight block cipher implementation on the WIA-FA hardware platforms and under the specific protocol requirements for factory automation.

Abbreviations

AES: Advanced Encryption Standard; DESL: Data Encryption Standard Lightweight; DESXL: XORed variant of DESL; DES: Data Encryption Standard; GEs: Gate equivalents; IOTs: Internet of Things; RFID: Radio-frequency identification; WSNs: Wireless sensor networks; WIA-F: Wireless Network for Industrial Automation for Factory Automation; WIA: Wireless Network for Industrial Automation; XTEA: Extensions to Tiny Encryption Algorithm; XXTEA: Corrected Block Tiny Encryption Algorithm

Funding

The work is partially supported by the following funding: the National Natural Science Foundation of China (NSFC), no. 61374200 as well as National Natural Science Foundation of China, Sino-Korea Cooperation Project no. 71661147005, and Ministry of Science and Technology Inter-Governmental International Scientific and Technological Innovation Cooperation Key Projects YS2017YFGH000571.

Authors' contributions

The first author conducted the experiments and wrote the first draft of the paper. Other co-authors helped to revise the paper and polished the paper. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Author details

[1]Key Laboratory of Networked Control Systems, Chinese Academy of Sciences, 110016, Shenyang, China. [2]Shenyang Institute of Automation, Chinese Academy of Sciences, 110016, Shenyang, China. [3]University of Chinese Academy of Sciences, 100049, Beijing, China. [4]Department of Computer Science, The University of Alabama, Tuscaloosa, 5487-02903 AL, USA.

Received: 5 January 2018 Accepted: 20 April 2018
Published online: 10 May 2018

References

1. Y Xiao, S Yu, K Wu, Q Ni, C Janecek, J Nordstad, Radio frequency identification: technologies, applications, and research issues. Wireless Commun. Mobile Comput. **7**, 457–472 (2007)
2. Y Xiao, X Shen, B Sun, L Cai, Security and privacy in RFID and applications in telemedicine. IEEE Commun. Mag. **44**, 64–72 (2006)
3. HEH Mustafa, X Zhu, Q Li, G Chen, Efficient median estimation for large-scale sensor RFID systems. Int. J. Sensor Netw. **12**, 171–183 (2012)
4. KT Nguyen, M Laurent, N Oualha, Survey on secure communication protocols for the Internet of Things. Ad Hoc Netw. **32**, 17–31 (2005)
5. S Xiong, L Tian, X Li, L Wang, Fault-tolerant topology evolution and analysis of sensing systems in IoT based on complex networks. Int. J. Sensor Netw. **18**, 22–31 (2005)
6. H Cheng, N Xiong, AV Vasilakos, L Yang, G Chen, Nodes organization for channel assignment with topology preservation in multi-radio wireless mesh networks. Ad Hoc Netw. **10**(5), 760–773 (2012)
7. H Cheng, Z Su, N Xiong, Y Xiao, Energy-efficient nodes scheduling algorithms for wireless sensor networks using Markov Random Field Model. Inform. Sci. **329**, 461–477 (2016)
8. M Faisal, AA Cardenas, Incomplete clustering of electricity consumption: an empirical analysis with industrial and residential datasets. Cyber-Physical Syst. **3**(1–4), 42–65 (2017)
9. Latré B, B Braem, I Moerman, C Blondia, P Demeester, A survey on wireless body area networks. Wireless Netw. **17**, 1–18 (2011)
10. G Anastasi, M Conti, M Di Francesco, A Passarella, Energy conservation in wireless sensor networks: a survey. Ad hoc Netw. **7**, 537–568 (2009)
11. J Liu, Y Xiao, Temporal accountability and anonymity in medical sensor networks. Mob. Netw Appl. **16**, 695–712 (2011)
12. C Liu, S Ghosal, Z Jiang, S Sarkar, An unsupervised anomaly detection approach using energy-based spatiotemporal graphical modeling. Cyber-Physical Syst. **3**(1-4), 66–102 (2017)
13. A Olteanu, Y Xiao, F Hu, B Sun, H Deng, A lightweight block cipher based on a multiple recursive generator for wireless sensor networks and RFID. Wireless Commun. Mobile Comput. **11**, 254–266 (2011)
14. M Cazorla, K Marquet, Minier M, in *Proceedings of the 2013 International Conference on Security and Cryptography (SECRYPT)*. Survey and benchmark of lightweight block ciphers for wireless sensor networks (IEEE, Reykjavik, 2013), pp. 1–6
15. B Sun, CC Li, K Wu, Y Xiao, A lightweight secure protocol for wireless sensor networks. Comput. Commun. **29**, 2556–256 (2006)
16. G Ferrari, F Cappelletti, R Raheli, A simple performance analysis of RFID networks with binary tree collision arbitration. Int. J. Sensor Netw. **4**, 194–208 (2008)
17. SE Sarma, SA Weis, DW Engels, in *Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems*. RFID systems and security and privacy implications (Springer, REDWOOD SHORES, 2002), pp. 454–469
18. K Finkenzeller, *In RFID handbook: fundamentals and applications in contactless smart cards, radio frequency identification and near-field communication, 1st Eds*. (John Wiley Sons Ltd, West Sussex, 2010)
19. A Juels, Weis SA, in *Proceedings of the 25th Annual International Cryptology Conference*. Authenticating pervasive devices with human protocols (Springer, Santa Barbara, 2005), pp. 293–308
20. JH Kong, LM Ang, KP Seng, A comprehensive survey of modern symmetric cryptographic solutions for resource constrained environments. J. Netw. Comput. Appl. **49**, 15–50 (2005)
21. J He, Z Xu, Authentication and search mechanism for diffusing RFID-sensor networks. Int. J. Sensor Netw. **14**, 211–217 (2013)
22. T Eisenbarth, Z Gong, T Güneysu, S Heyse, S Indesteege, S Kerckhof, FX Standaert, in *Proceedings of the 5th International Conference on*

Cryptology in Africa, Ifrane, Morocco. Compact implementation and performance evaluation of block ciphers in ATtiny devices (Ifrane, Springer, 2012), pp. 172–187
23. A Olteanu, Y Xiao, F Hu, B Sun, H Deng, A lightweight block cipher based on a multiple recursive generator for wireless sensor networks and RFID. Wireless Commun. Mobile Comput. **11**, 254–266 (2011)
24. B Sun, Y Xiao, CC Li, HH Chen, TA Yang, Security co-existence of wireless sensor networks and RFID for pervasive computing. Comput. Commun. **31**, 4294–4303 (2008)
25. JP Kaps, in *Proceedings of the 9th Annual International Conference on Cryptology in India*. Chai-Tea, Cryptographic hardware implementations of xTEA (Springer, Kharagpur, 2008), pp. 363–375
26. E Yarrkov, *Cryptanalysis of XXTEA. International Association for Cryptologic Research (IACR) Cryptology EPrint Archive*, (2010)
27. G Leander, C Paar, A Poschmann, K Schramm, in *Proceedings of the 14th International Workshop on Fast Software Encryption*. New lightweight DES variants (Springer, Luxembourg, 2007), pp. 196–210
28. A Poschmann, G Leander, K Schramm, C Paar, in *Proceedings of the IEEE International Symposiumon Circuits and Systems*. New light-weight crypto algorithms for RFID (IEEE, New Orleans, 2007), pp. 1843–1846
29. M Izadi, B Sadeghiyan, SS Sadeghian, HA Khanooki, in *Proceedings of the 8th International Conference on Cryptology and Network Security*. MIBS: a new lightweight block cipher (Springer, Kanazawa, 2009), pp. 334–348
30. A Bay, J Nakahara Jr, S Vaudenay, in *Proceedings of the 9th International Conference on Cryptology and Network Security*. Cryptanalysis of reduced-round MIBS block cipher (Springer, Kuala Lumpur, 2010), pp. 1–19
31. C Canniere De, O Dunkelman, M Knezevic, in *Proceedings of the 11th International Workshop on Cryptographic Hardware and Embedded Systems*. KATAN and KTANTAN–a family of small and efficient hardware-oriented block ciphers (Springer, Lausanne, 2009), pp. 272–288
32. AA Priyanka, SK Pal, A survey of cryptanalytic attacks on lightweight block ciphers. Int. J. Comput. Sci. Inf. Technol. Secur. (IJCSITS). **2**, 472–481 (2012)
33. T Suzaki, K Minematsu, S Morioka, Kobayashi E, in *Proceedings of the ECRYPTWorkshop on Lightweight Cryptography*. TWINE: a lightweight, versatile block cipher (Springer, Belgium, 2011)
34. M Coban, F Karakoc, O Boztas, in *Proceedings of the 11th International Conference on Cryptology and Network Security*. Biclique cryptanalysis of TWINE (Springer, Berlin, 2012), pp. 43–55
35. G Bansod, N Pisharoty, A Patil, PICO: an ultra lightweight and low power encryption design for ubiquitous computing. Defence Sci. J. **66**, 259–265 (2016)
36. Y Xiao, HH Chen, X Du, M Guizani, Stream-based cipher feedback mode in wireless error channel. IEEE Trans. Wireless Commun. **8**, 622–626 (2009)
37. X Liang, Y Xiao, S Ozdemir, AV Vasilakos, H Deng, Cipher feedback mode under go-back-N and selective-reject protocols in error channels. Secur. Commun. Netw. **6**, 942–954 (2013)
38. A Olteanu, Y Xiao, in *Proceedings of the 2009 IEEE International Conference on Communications (ICC 2009)*. Fragmentation and AES encryption overhead in very high-speed wireless LANs (IEEE, Dresden, 2009), pp. 575–579
39. A Olteanu, Y Xiao, Security overhead and performance for aggregation with fragment retransmission (AFR) in very high-speed wireless 802.11 LANs. IEEE Trans. Wireless Commun. **9**, 218–226 (2010)
40. Olteanu A, Y Xiao, Y Zhang, Optimization between AES security and performance for IEEE 802.15.3 WPAN. IEEE Trans. Wireless Commun. **9**, 6030–6037 (2009)
41. J Shen, S Chang, J Shen, Q Liu, X Sun, A lightweight multi-layer authentication protocol for wireless body area networks. Future Generation Comput.Syst (2016). https://doi.org/10.1016/j.future.2016.11.033
42. Z Zhou, QMJ Wu, F Huang, X Xingming Sun, Fast and accurate near-duplicate image elimination for visual sensor networks. Int. J. Distributed Sensor Netw. **13**(2) (2017). https://doi.org/10.1177/1550147717694172
43. J Zhang, J Tang, T Wang, F Chen, Energy-efficient data-gathering rendezvous algorithms with mobile sinks for wireless sensor networks. Int.J. Sensor Netw. **23**(4), 248–257 (2017). https://doi.org/10.1504/IJSNET.2017.10004216
44. Y Sun, F Gu, Compressive sensing of piezoelectric sensor response signal for phased array structural health monitoring. Int. J. Sensor Netw. **23**(4), 258–264 (2017). https://doi.org/10.1504/IJSNET.2017.10004214
45. X Chen, S Chen, Y Wu, Coverless information hiding method based on the Chinese character encoding. J. Internet Technol. **18**(2), 313–320 (2017). https://doi.org/10.6138/JIT.2017.18.2.20160815
46. Y Zhang, X Sun, B Wang, Efficient algorithm for k-barrier coverage based on integer linear programming. China Commun. **13**(7), 16–23 (2016). https://doi.org/10.1109/CC.2016.7559071
47. B Wang, X Gu, Ma L, S Yan, Temperature error correction based on BP neural network in meteorological WSN. Int. J. Sensor Netw. **23**(4), 265–278 (2017). https://doi.org/10.1504/IJSNET.2017.083532
48. Z Qu, J Keeney, S Robitzsch, F Zaman, X Wang, Multilevel pattern mining architecture for automatic network monitoring in heterogeneous wireless communication networks. China Commun. **13**(7), 108–116 (2016). https://doi.org/10.1109/CC.2016.7559082
49. H Zhang, Shi Cheng P L, J Chen, Optimal DoS attack scheduling in wireless networked control system. IEEE Trans. Control Syst. Technol. **24**, 843–852 (2016)
50. IEC PAS 62948. In industrial networks—wireless communication network and communication profiles - WIA-FA, 1st Eds, 2015; Available online: http://www.iec.ch
51. Z Gong, S Nikova, Law YW, in *Proceedings of the 7th Workshop on RFID Security and Privacy (RFIDSec)*. KLEIN: a new family of lightweight block ciphers (Springer, Amherst, 2011), pp. 1–18
52. W Wu, L Zhang, in *Proceedings of the 9th International Conference on Applied Cryptography and Network Security (ACNS)*. LBlock: a lightweight block cipher (Springer, SPAIN, 2011), pp. 327–344
53. A Bogdanov, LR Knudsen, G Leander, C Paar, A Poschmann, MJB Robshaw, Y Seurin, Vikkelsoe C, in *Proceedings of the 9th International Workshop on Cryptographic Hardware and Embedded Systems*. PRESENT: an ultra-lightweight block cipher (Springer, Vienna, 2007), pp. 450–466
54. D Hong, J Sung, S Hong, J Lim, S Lee, B Koo, H Kim, in *Proceedings of the 8th International Workshop on Cryptographic Hardware and Embedded Systems*. HIGHT: a new block cipher suitable for low-resource device (Springer, Yokohama, 2006), pp. 46–59
55. K Shibutani, T Isobe, H Hiwatari, A Mitsuda, T Akishita, Shirai T, in *Proceedings of the 13th International Workshop on Cryptographic Hardware and Embedded Systems*. Piccolo: an ultra-lightweight blockcipher (Springer, Nara, 2011), pp. 342–357
56. R Beaulieu, D Shors, J Sntith, S Treatrnan-Cark, B Weeks, L Wingers, in *Proceedings of the IT Professional Conference (IT Pro)*. The simon and speck families of lightweight block ciphers (IEEE, Gaithersburg, 2014)
57. Pub NF, In 197: advanced encryption standard (AES). Federal Inf. Process. Standards Publication. **197**, 441–0311 (2001)
58. Y Xiao, B Sun, HH Chen, in *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM 06)*. Performance analysis of advanced encryption standard (AES) (IEEE, San Francisco, 2006), pp. 1–5
59. Y Xiao, HH Chen, B Sun, R Wang, S Sethi, MAC security and security overhead analysis in the IEEE 802.15.4 wireless sensor networks. EURASIP J. Wireless Commun. Netw. **1**, 1–12 (2006)
60. S Kerckhof, F Durvaux, C Hocquet, D Bol, FX Standaert, in *Proceedings of the 14th International Workshop on Cryptographic Hardware and Embedded Systems*. Towards green cryptography: a comparison of lightweight ciphers from the energy viewpoint (Springer, Leuven, 2012), pp. 390–407
61. BJ Mohd, T Hayajneh, AV Vasilakos, A survey on lightweight block ciphers for low-resource devices: comparative study and open issues. J. Netw. Comput. Appl. **58**, 73–93 (2005)
62. C Manifavas, G Hatzivasilis, K Fysarakis, K Rantos, in *Proceedings of the 8th Data Privacy Management International Workshop (DPM)*. Lightweight cryptography for embedded systems–a comparative analysis (Springer, Egham, 2014), pp. 333–349
63. JCH Castro, JM Sierra, A Seznec, A Izquierdo, A Ribagorda, The strict avalanche criterion randomness test. Math. Comput. Simul. **68**, 1–7 (2005)
64. R Beaulieu, D Shors, J Sntith, S Treatrnan-Cark, B Weeks, L Wingers, in *Proceedings of the 52nd ACM/EDAC/IEEE Design Automation Conference (DAC)*. The SIMON and SPECK lightweight block ciphers (IEEE, New York, 2015), pp. 1–6