Al-Rafidain Journal of Computer Sciences and Mathematics (RJCM)

www.csmj.mosuljournals.com

Software Implementation Solutions of A Lightweight Block Cipher to Secure Restricted IoT Environment: A Review

Ruah Mouad Alyas Al_Azzawi[1]*, Sufyan Salim Mahmood Al-Dabbagh[2]

Computer Science Department, College of Computer Science and Mathematics, University of Mosul, Mosul, Iraq
**Corresponding author. Email: ruaa.moayad@uomosul.edu.iq[1]*

Article information

Article history:
Received : 30/6/2022
Accepted : 10/10/2022
Available online :

Abstract

With the development of the Internet of Things (IoT) technology, IoT devices are integrated into many of our daily lives, including industrial, security, medical, and personal applications. Many violations of IoT safety have appeared due to the critical physical infrastructure, and network vulnerabilities. Considering the nature of the restricted and limited resources of these devices in terms of size, capacity, and energy, Security is becoming increasingly important. Lightweight cryptography is one of the directions that offer security solutions in resource-constrained environments such as Radio-frequency identification (RFID) and wireless sensor network (WSN).This paper discusses the security issues of these resource-constrained IoT devices and reviews the most prominent Lightweight Bock Cipher suitable for software implementation. Through studying the specifications and the inner structure for each cipher and their implementation of the performance evaluation on some kind of platform, we provide a design strategies guideline for cryptographic developers to design improved Lightweight Block cipher solutions and compact software implementation for resource-constrained environments.

Keywords:

Block Ciphers, Internet of Things devices Security, Lightweight cryptography, Performance Evaluation, Resource-constrained devices.

Correspondence:
Author: Ruah Mouad Alyas Al_Azzawi
Email: ruaa.moayad@uomosul.edu.iq

1. INTRODUCTION

Because of their omnipresent nature, security plays a significant role in authenticating information in communication systems and other applications such as IoT applications. IoT technology is currently being employed in a variety of applications, such as smart infrastructure (smart homes, smart cities, and smart grid), wearable technology, and smart automobiles, with numerous uses in the automotive system and elsewhere.

By the end of 2020, it is estimated that more than 18 billion IoT devices would be on the market and connected through the cloud, with more than half of them for industrial uses [1]. As technology connects a lot of devices through the Internet, hacking them can have a big loss, such as losing sensitive personal and economic information, when user's lack of knowledge about how to work with these devices and the potential risks to personal information due to misuse.

Users need to keep their data private when using these applications. This led to a change in the trend in adopting safety as a basic thing in the manufacture of these devices, especially if they are used in sensitive applications (such as identification, credit cards, personal and confidential data for patients, etc.). Ubiquitous computing with large networks of resource constrained IoT devices, have extremely tight cost constraints over time, Moore's law can expect the speed and capability of our computers to increase every two years, though the cost of computers is halved, which will increasingly enable such applications[2].The demand for cryptographic components is significant and growing since many of these applications will process sensitive information monitoring or biometric data. Confidentiality, Integrity and Authentication (CIA) are the three basic requirements of securing any system.

Encryption is one of the most effective methods for providing an End-to-end security. The authors in [3] described the main IoT security mechanisms, to meet confidentiality nodes are encrypted to achieve end to End-to-end security. Security mechanisms like Authentication techniques, Access control and the new technology block chain and software defined network (SDN) another solution to ensure the privacy of the end users, data, infrastructures and all devices of the IoT system. When dealing with low-power and restricted devices, encryption research focuses on finding a balance between security concerns and low-cost encryption.

2. Security Challenges of Resource-Constrained devices in IoT applications:

1. Privacy and trust issues: private information must be kept private, protected, and guaranteed to be sent only to the legal person or device. Manufacturers and service providers do not prioritize providing security and privacy in their products.
2. External threats, such as eavesdropping, manipulation, DoS attacks, Man-in-the-Middle attacks, phishing attacks, Side channel attack and code injection. IoT platforms with embedded devices are more susceptible to these threats. Physical attacks on these devices make unprotected data easily accessible. Several security threats, attacks, techniques, countermeasures, and solutions for IoT environments are reviewed in [64][65].
3. To communicate with smart devices, various communication protocols, Zigbee, Bluetooth Low Energy, 6LoWPAN, CoAp are employed. Sinkhole and many attacks appear to be one of several weaknesses. Security characteristics and difficulties of the most popular wireless communication protocols for IoT applications in smart cities are describe in [65].
4. Different devices connected in the IoT ecosystem range from high-resource devices like servers, personal computers, and smartphones to low-resource devices like sensor nodes, sensor nodes, RFID tags, and wireless sensor networks (WSN), among others. Because of their limited memory, power, and processing capabilities, the security of these confined objects is an issue, and traditional solutions are not always available for them.

3. Cryptography Solutions Challenges implementation in Resource-Constrained IoT Devices

Cryptographic algorithms are employed to ensure the secrecy, integrity, authentication, and authorization of data traveling via resource-constrained IoT devices, as well as to safeguard data stored or transiting over the network. Figure 1 illustrates the role of the cryptographic techniques to prevent attacker from reaching the IoT data and tampering it. Due to resource limits, implementing standard cryptography in these IoT devices is difficult:

- heavy and complex mathematical operation
- operations use huge memory space
- Traditional cryptography is expensive to implement on low-resource devices (circuit size) which impose challenges on software design implementation. To overcome these difficulties, lightweight ciphers were introduced [2, 3, 4].

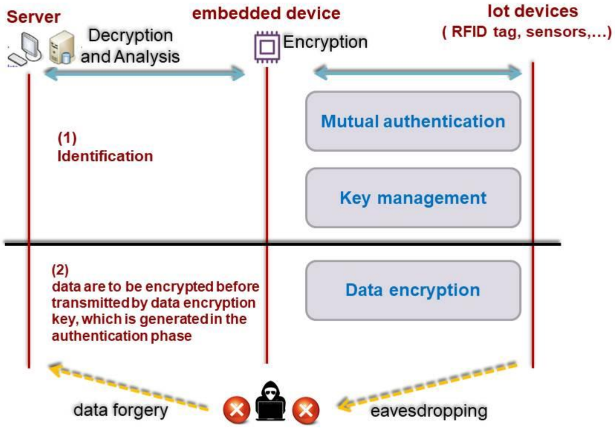

Fig. 1 Security for IoT Based on Encryption

4. Lightweight cryptography (LWC)

Lightweight cryptography is a branch of a current cryptographic technique aimed at providing security solutions for devices such as mobile phones, RFID tags, sensor networks, smart cards, and IoT devices. [5].

In 2015, the National Institute of Standards and Technology (NIST) announced a contest for a suitable lightweight cryptographic algorithm that might be used in resource-constrained environment [6] .NIST claimed that symmetric ciphers may be used to provide high-level security in low-end devices such as RFID Tags, Motes, Smartcards, Industrial Sensors, Wireless Sensors, Mobile or User Equipment, Healthcare Devices (like Hubs), and other battery-powered devices like Wearable[6].Three criteria were used to accept or reject algorithms during the evaluation process: cryptographic security, performance, and implementation cost. Other features were examined in the second candidate, including functionality, underlying components, design methods, and supported key and tag sizes [6].Lightweight Cryptography Working Group was established in 2013 by CRYPTREC and they published a comprehensive technical report about LWC in 2017.

To secure resource-constrained devices, many algorithms have been presented in this domain, which either software-based or hardware-based implementation of lightweight ciphers [7]. These ciphers aim to minimize the overall implementation costs for cryptographic primitives that are hardware-oriented as well as software-oriented in terms of various aspects, such as:(Number of rounds, memory, key size, power consumption, throughput, and Gate Equivalence (GE)). This is accomplished by focusing on:

- Hardware costs (power consumption, physical area (GE), and energy consumption) are all being reduced.
- Improved software efficiency (memory consumption, processing power, throughput, and latency).

New algorithms are introduced in this literature that are based on either modified versions of well-known cryptographic algorithms by adaptive ones for the constrained environment or designing a new cipher algorithm that meets the requirements to secure the resource-constrained devices.

There are a variety of LWC algorithms available nowadays, authors of a number of scholarly publications [8].In [2] the authors presented a comparison between many adaptive algorithms for embedded systems that were designed for good hardware performance. In [9] the authors proposed a new block cipher, Lightweight Data Encryption Standard (DESL), which is a modified version of the DES algorithm. They use a single S-Box instead of eight S-boxes in DES to reduce the cost of implementation. Designing a new lightweight cryptographic algorithm, on the other hand, had a wide branch to secure constrained devices, with a large variety of algorithms.

5. Lightweight Block Cipher

In recent years, many strategies with highly limited applications have been created. Lightweight block ciphers, in particular, can get beyond the limitations of these applications. The choice of a lightweight block cipher is crucial since it affects the system's cost, area, speed, latency, and bandwidth requirements. Several considerations are made when building a new Lightweight block cipher algorithm to lower the cost of resource consumption:

5.1 Block Cipher inner structure:

Based on their inner structure, block ciphers can be classified into: Substitution Permutation Networks (SPN), Feistel Networks, Generalized Feistel network (GFN), Add-Rotate-XOR (ARX), and hybrid. In SPN, the plaintext is transformed and prepared for the next round by using a series of successive substitutions and permutation boxes. SPN provides a higher level of protection, but it also consumes more resources. Feistel networks use a round function to conduct a diffusion function on half of the data in each block. Although many applications do not require decryption, it uses a smaller round function to provide both encryption and decryption at a low cost. GFN takes a data block and splits it into sub-blocks, applying the Feistel functions to each pair of sub-blocks. GFN encrypts and decrypts using the same round function, making it a good choice for low-cost hardware implementation. With no S-boxes, ARXs use simple operations like addition, rotation, and XORs. Compared to SPN and Feistel ciphers, ARX's security properties have not been thoroughly investigated, yet, they produce small and rapid implementations. Hybrid ciphers combine the three types of ciphers discussed above in order to improve various efficiency measures.

5.2 Targeted Implementation Environment (Hardware or Software)

Depending on the implementation Environment Lightweight cryptographic algorithms can be classified to hardware or software implementation. The main goal of hardware implementation is to achieve minimal gate equivalent by reducing the number of logic gates required. This lowers the cost and reduces the amount of power consumption. Hardware implementations are better suited to ultra-constrained devices like 4-bit microcontrollers that execute specified functions. Software implementations target to small memory consumption, processing power, and throughput (bytes per cycle), these design require a microprocessor to operate.

Cryptographic libraries for embedded devices include Software implementations. In comparison to hardware implementations, their key advantage is portability. Three software implementations on different restricted environment (8-bit) AVR processor, (6-bit) MSP processor and (32-bit) ARM processor are introduced by FELICS framework [10], to evaluate the performance of lightweight block or stream ciphers in terms of implementation size, RAM utilization, and time to complete a given operation. Table 1 presents a basic comparison of some common lightweight block ciphers, we are primarily interested in software implementation design, and display the structure of the top selected lightweight block ciphers for software implementation design. The following are the details of all of the lightweight block ciphers that were chosen for software implementation design:

- **TEA / XTEA** [11][12]

Wheeler and Needham presented the Tiny Encryption Algorithm (TEA) in 1994. It has Feistel structure and small amount of code and can be easily integrated into embedded systems. XTEA is an improved version of TEA to overcome the discovered weakness in it [12], it has a complex key-schedule than TEA , and also based on simple F-function composed of left and right shifts operations, XORs and additions.

- **KASUMI /MISTY1**[13]

MISTY1was presented by M Matsui in 1997.it has Feistel structure ,KASUMI has Feistel structure and it is equivalent to MISTY1 in 8 rounds, with the exception that its key schedule rotates the bits of the master key and XORs round constants. It is used in the worldwide system for mobile communication (GSM), UMTS, and GPRS for security purposes. [14].

- **AES** [15]

The Advance Encryption Standard (AES) was created by Vincent Rijmen and Joan Daemen in 1998 and was adopted as the encryption standard by NIST in 2001. It accepts 128 bits of plain text as input and produces 128 bits of encrypted cipher text as output. It calculates all the round keys from the original key using a Key Schedule method. The number of rounds is determined by the key length: 10 for a 128-bit key, 12 for a 192-bit key, and 14 for a 256-bit key. Instead of working with bits at a time, it works with bytes of data. The input block size is 128 bits (or 16 bytes), and the cipher state is displayed as 4*4 matrixes, with four operations applied in the following order: Substitute Bytes (does the substitution), ShiftRows (does the permutation), and MixColumns (does the permutation), and Add Round key (does the permutation). The four operations for decryption will be: Add a round key, and then reverse MixColumns, ShiftRows, and Inverse SubByte. Despite the fact that it is not a lightweight encryption, many IoT devices use this technique.

- **Camellia** [16]

Camellia was designed by Nippon Telegraph and Telephone Corporation and Mitsubishi Electric Corporation in 2000. It is an ISO/IEC, IETF, NESSIE and CRYPTREC recognized cipher and offers a similar level of security as AES .it has Feistel structure with two round variants, 18 rounds (when using 128 bit keys) or 24 rounds (when using 192 or 256 bit keys).

- **HIGHT** [17, 18]

Hong et al. presented this encryption in 2006. It has a GFS structure based on ARX. Its main operations are XOR, addition mod 28 and left bitwise rotation. WhiteningKey Generation (create 8 whitening key bytes used in the first and last rounds) and SubkeyGeneration are the two algorithms that make up the key schedule (generates 128 subkey bytes). The authors of [18] presented a software and hardware implementation of the HIGHT block cipher for resource-constrained devices (8-bit AVR and 32-bit ARM Cortex-M3) and ASICs.

- **SEA** [19]

Francois-Xavier et al. presented this encryption in 2006. It has a Feistel structure that can be used in software on an 8-bit processor. Its F-function is made up of basic operations: Bitwise XOR, apply 3x3 S-box, word rotation, bit rotation and Addition modulo 2b, this enables for quick evaluation, minimal memory usage, and short code size.

- **CLEFIA** [20]

This cipher is proposed by Sony in 2007 and presented as standardization in ISO/IEC 29192. It has type-2 GFN structure, The 128 bit (16 bytes) plaintext input P0 to P15 is grouped in 4 byte words. It uses a simpler key scheduler and small F-functions, with small S-Boxes and basic permutations. CLEFIA uses whitening keys WK0 to WK3 at the start and end of encryption.

- **KLEIN** [21]

KLEIN has been by Zheng Gong et al. in 2011, it is based on SPN, for software efficiency on 8-bit processors, and preferred byte-oriented matrix multiplication operations. Each round has four layers in order: AddRoundKey, SubNibbles, RotateNibbles, and MixNibbles. The author of [22] chose the KLEIN cipher as the most lightweight security solution to test in an IoHT environment.

- **LBlock** [23]

This cipher proposed was by Wu and Zhang in 2011, it has Feistel Network structure and has an efficient software implementation on 8-bit microcontrollers .Its round function consists of substitution layer using 4-bit S-boxes (8 Sboxes applied in parallel) and permutation layer (32-bit permutations with shift operations).

- **LED** [24]

The Lightweight Encryption Device (LED) was proposed by Guoin 2011,it has SPN structure, its operation is similar to an AES-like design ,each round applies 4 functions: AddConstants, SubCells (applies a 4-bit Sbox Present cipher),ShiftRows and MixColumnsSerial(using Maximum Distance Separable (MDS)).

- **TWINE** [25]

This cipher was presented by Tomoyasu Suzaki et al. in 2011. It has Type-2 GFS with 16 of 4-bits branches. twine has efficient software implementation on various platforms, Its F-function consist of only a subkey addition and a nonlinear substitution layer using single 4-bit S-box that acts on nibbles with repetition 8 times every round , and a diffusion layer that permutes the blocks of 4 bits.

- **SPECK and SIMON** [26, 27]

SPECK and SIMON have been presented by The U.S. National Security Agency (NSA) in2013. SPECK is ARX and performs 22, 23, 26, 27, 28, 29, 32, 33 and 34 iterations. Each round consisting of: Bitwise XOR, Addition modulo 2^n and Left and Right circular shift, make it suited to software implementations more than Simon.

Simon uses a Feistel structure with simple arithmetic and logic operations, its round function consist of left circular shifts, bitwise XOR and bitwise AND. If the block size consist of 2n-bits and a key size of mn-bits then it represented as 2n/mn.

- **ITUBEE**[28]

This cipher was presented by F Karakoç et al. in 2013.it has Feistel structure with no key schedule making it suitable for 8-bit software-based platforms with limited resources. It Insert round keys between two round functions F to strengthen the cipher against related key attacks.

- **Chaskey** [29]

Nicky Mouha et al. presented the Chaskey cipher for 32-bit microcontrollers in 2014. It's ARX, with a permutation-based MAC technique based on an Even-Mansour block cipher as the foundation. The XOR with state method is used to generate the keys. Because key updating consists of two shifts and two XORs for two subkeys, there is no key schedule.

- **Fantomas [**30]

Vincent Grosso proposed this cipher in 2014. It uses LS-designs, which combine L-boxes (look-up tables) and bit-slice S-boxes. On 8-bit MCUs, Fantomas has a good implementation.

- **Robin** [30, 31]

Robin was proposed by Vincent Grosso 2014, it has SPN structure and similar to Fantomas, but uses involutions on its L-Box and its S-Box (8×8 bits S-Box and a 16×16 bits L-Box) to be used for decryption and encryption.

- **FeW** [32]

This cipher was proposed by Kumar, et al. in 2014.It has based on Feistel-M structure consist of two Feistel branches of 4-branch generalized Feistel structure to improve security against cryptographic attacks. It utilizes Humminbird-2's S-box and imitates the key expansion process from the PRESENT.

- **Pride**[33]

Albrecht et al proposed this cipher in 2014, it has an SPN structure and is easy to implement in software on 8-bit microcontrollers. It has a strong linear layer separated into three sub-layers and a bit-sliced S-box. The 128-bit master key is split into two key, k0 and k1, which are used to encrypt data. Pre-whitening and post-whitening are handled by k0 (64 bits), whereas the subkey for each round is handled by k1 (64 bits).

- **RECTANGLE**[34]

RECTANGLE proposed by Zhang et al in 2015, it is an ultra-lightweight block SPN cipher, with a substitution layer consists of 4-bit S-boxes connected in parallel and a permutation layer executed in 3 rotations. There are three operations in each round: 1. SubColumn, 2.AddRoundkey (using Bitwise XOR with round key), and 3.ShiftRow (each row is rotated left over different offsets), which uses bit-slice techniques to obtain a fast software speed.

- **SIMECK**[35]

Gangqiang Yang et al. first proposed this cipher in 2015. SIMECK is a Feistel block cipher that combines the best design elements of SIMON and SPECK block ciphers. It employs ARX operations to encrypt or decode 2n-bit message blocks utilizing a 4n-bit key and 2n-bit message blocks. Changes in the rotations and key scheduling enable for better hardware and software implementation. Efficient implementation methods of Simeck were proposed in [36, 37] these proposed methods can be adapted in IoT application.

- **RoadRunneR**[38]

RoadRunneR is presented by Adnan Baysal and Sühap Sahin in 2015, it is Feistel bit-slice block cipher that is targeted for software implementations on CPUs with an 8-bit architecture. It follows LS-Design in which the cipher is composed of S-Boxes that follow the bit slice and L-Boxes (linear P-Boxes).It uses 3 keys per round plus 2 whitening keys one in the beginning and another at the end to XOR with the block. In encryption the 64 bit block is divided into two32bit parts, the left part is XORed with whiteningkey at the beginning and end of the encryption.

- **SPARX** [39, 40]

In 2016, Daniel Dinu et al. presented the SPARX cipher, which is built on the ARX structure and enhance its security with an SPN structure. Rather than storing Speckey S-Box in RAM, it constructs it using simple procedures. They suggest a new method called "Long Trail Strategy" (LTS) in place of "wide trail design strategy" (WTS), which advises the use of large and computationally expensive S-boxes combined with light linear layers termed Long Trail Argument.

- **ANU** [41]

ANU was presented by G. Bansod et al. in 2016 as an ultra-lightweight block cipher with Feistel- network structure. The key scheduling is motivated by the key schedule of PRESENT cipher. The round function has two operations in which F1 (left circular shift by 3 bit) and F2 (right circular shift by 8 bit). F1 output is applied to the nonlinear layer S-box then XORed with the LSB 32 bit data resulting in FX which is XORed with F2 and with round key. ANU is well-suited to applications with tight constraints, such as IoT.

- **PICO** [42]

This cipher was developed by Bansod et al in 2016, it is ultra-light SPN block cipher. It has three operations involved in encryption process: AddRoundkey, SubColumn and the Bit_Shuffle. The PICO cipher key schedule is based on the SPECK cipher key scheduling architecture, it uses key of 128 bit to extract 33 subkeys k0-k32 of size 64 bits and K32 is used for post whitening key.

- **SKINNY** [43]

This cipher proposed was by Beierle, et al. in 2016. SKINNY family have SPN structure .It employs three key-length possibilities of n bits, 2n bits, or 3n bits, with n being the block size (64 or 128 bits). The number of rounds varies from 32 to 56 depending on the block and encryption key size. It includes a light key scheduling and light diffusion layer.

- **SIT** [44]

This cipher is proposed by Muhammad Usman et al.in 2017. SIT (Secure IoT) is hybrid approach based on combining Feistel with SPN structure. Encryption process is composed of logical operations, left shifting, swapping and substitution. 5 different keys are used for 5 rounds encryption to improve energy efficiency.

- **LiCi** [45]

Patil et al. proposed this cipher in 2017, and it has a Feistel structure. The MSB of the input plaintext is sent into 8 S-boxes for replacement after the 64-bit input is separated into two pieces, each of which includes 32 bits. It uses 4-bit S-boxes with simple operations, XOR, left and right circular shift for encryption process. LiCi key scheduling algorithm is inspired by PRESENT cipher.

- **CHAM** [46]

This cipher was presented by Koo et al. in ICISC 2017. The CHAM family consists of three cipher standards, CHAM-64/128, CHAM-128/128 with 80 rounds and CHAM-128/256 with 96 rounds. It is generalized 4-branch Feistel structure based on ARX operations.

- **GIFT** [47]

This cipher was proposed by Banik et al. in 2017**,** it has SPN structure Based on PRESENT cipher and overcome the weakness in it. GIFT has two versions GIFT-64 with 28 rounds and GIFT-128 with 40 rounds according to the block size and with key of 128 bit. The round function consist of 3 subfunctions named SubCells (apply 4-bit Sbox), PermBits and AddRoundKey.

- **BRIGHT** [48]

This cipher is proposed by Sehrawat and Gill in 2019, it is GFN-based based on 4-branch block cipher for resource-constrained IoT applications devices. The number of rounds different from 32 to 37 depending on the cipher block and encryption key sizes. The block size is 64-bit or 128-bit with an encryption key size ranging from 80 to 256 bits. It uses three layers, first pre-key whitening and, for each round applied second layer which perform ARX operations, and third layer perform round permutation.

- **NLCA**[59]

This algorithm is proposed by Thabit et al. in 2021, it is structure based on combination between FN and SPN for enhancing data transmission security in cloud services. XOR, XNOR, F functions, swaps, and other transformation are used in each round. NLCA performance including execution time and lower memory usage was evaluated against some popular cryptographic algorithms, including DES, AES, HIGHT, Blowfish, and LED, utilizing a variety of parameters in the same cloud environment.

TABLE 1. Cryptographic properties of the selected Lightweight Block Ciphers

Algorithm	*Year*	*Key size in bits*	*Block size in bits*	*Rounds*	*Structure*	*Target Environment*
TEA / XTEA	1994	128	64	64	Feistel	S.W
KASUMI/ MISTY1	1997	128	64	8	Feistel	H.W / S.W
AES	1998	128	128, 192, 256	10, 12, 14	SPN	H.W / S.W
Camellia	2000	128, 192, 256	128	18, 24	Feistel	H.W / S.W
HIGHT	2006	128	64	32	GFN +ARX	H.W / S.W
SEA	2006	96	96	Variable	Feistel	H.W / S.W
CLEFIA	2007	128, 192, 256	128	18, 22, 26	GFN	H.W / S.W
KLEIN	2011	64, 80, 96	64	12, 16, 20	SPN	H.W / S.W
LBlock	2011	80	64	32	Feistel	H.W / S.W
LED	2011	64, 80, 128	64	32, 48	SPN	H.W / S.W
TWINE	2011	80, 128	64	36	GFN	H.W / S.W
SIMON	2013	64, 72 ,96,128, 144, 192 , 256	32 , 48,64, 96,128	32, 36, 42,44,52, 54,68,69,72	Feistel	H.W / S.W
SPECK	2013	64, 72,96,128,144,192,256	32,48,64,96,128	22, 23, 26 ,27, 28, 29, 32, 33,34	ARX	S.W
ITUBEE	2013	80	80	20	Feistel	S.W
Chaskey	2014	128	128	8	ARX	S.W
Fantomas	2014	128	128	12	SPN	S.W
Pride	2014	128	64	20	SPN	H.W / S.W

FeW	2014	80, 128	64	32	GFN+ SPN	S.W
Robin	2014	128	128	16	SPN	S.W
RECTANGLE	2015	80, 128	64	25	SPN	H.W / S.W
RoadRunneR	2015	80, 128	64	10, 12	Feistel	S.W
SIMECK	2015	64, 96, 128	32, 48,64	32, 36,44	Feistel	H.W / S.W
SPARX and LAX	2016	128, 256	64,128	24,32 ,40	SPN+ARX	S.W
ANU	2016	80, 128	64	25	Feistel	H.W / S.W
PICO	2016	128	64	32	SPN	H.W / S.W
SKINNY	2016	64, 128, 192, 256,384	64,128	32 ,36, 40, 48,56	SPN	H.W / S.W
SIT	2017	64	64	5	Feistel+SPN	H.W / S.W
LiCi	2017	128	64	31	Feistel	H.W / S.W
CHAM	2017	128, 256	64, 128	64 ,128	GFN +ARX	H.W / S.W
GIFT	2017	128	64, 128	28, 40	SPN	H.W / S.W
BRIGHT	2019	80, 96, 128,192, 256	64 ,128	32, 33, 34 35 ,36 ,37	GFN	S.W
NLCA	2021	128	128	10, 20	FN+SPN	S.W

6. Performance Evaluation (Hardware and Software Performance Metrics)

Several measures based on hardware implementations or software implementations are offered to evaluate the performance of lightweight ciphers. The best encryption is one that provides a convenient level of security while balancing performance and cost concerns. This section summarizes the specifics of these measures. Some metrics are common whereas some are restricted to the H.W implementations (e.g. CMOS technology and GE metric) and others to S.W implementations (e.g. RAM size, ROM size).

6.1 Lightweight Ciphers Performance Evaluation Metrics:

Hardware technology: related to the CMOS technology that used to implement the lightweight cipher and there occupied circuit area which is measured in µm. 0.13 and 0.18 µm are the most technologies used in the lightweight cryptography research. Gate Equivalent (GE) metric is used to described the complication and the area occupied by the hardware implementations, this area represents the physical area required to run the cipher on a board measured in µm2 whereas(1GE = 2 input-NAND Gate).

Execution time: The execution time is measured by the number of clock cycles needed to complete each of a block cipher's operations (encryption, decryption, and key schedule encryption and key schedule decryption) for one data block. The executing time of the program is measure in milliseconds (ms) or seconds (s) .It can be calculated by the fraction of the amount of cycles to the frequency. In software implementation, this time is measured by resultant of the end time of operation subtracting the start time of operation.

Throughput: represented the cipher’s encryption operations and decryption operations obtained at a certain frequency. It measurable in bytes each CPU cycle. In hardware implementation, Throughput measure the plaintext processed per time unit (bits per second) at 100 KHz frequency, whereas in software implementation, Throughput represent the average amount of plaintext processed per CPU clock cycle at 4 MHz frequency.In software implementation, Throughput (bytes/ms) is measured by resultant of Data (in bytes) division by execution time depending on the processor’s frequency [52].

Cycles: It measured processor’s performance by calculating no. of the clock cycles used to calculate and read out the cipher text in hertz (Hz), megahertz (MHz) and gigahertz (GHz).

Latency: in Hardware performance represent the time to produce the cipher from the plain. While in software performance represent the number of clock cycles required to encrypt/decrypt a single block’s plain text/cipher text.

RAM/ROM Memory Requirements: it is calculated in KB, RAM presents the required byte to store intermediate values that used in operation .ROM is the required byte used to store the code size of the cipher and static data (key, S-box).

Efficiency: it is a trade-off between performance and implementation size. The higher metric is the better. For hardware implementations, Efficiency is calculated by the formula [63]:

$$\textit{Hardware Efficiency (Kbps/KGE)} = \textit{Throughput [Kbps]} * (\textit{Complexity [KGE]})^{-1} \quad (1)$$

Here, complexity in KGE which is the value of the physical area

$$\textit{Software Efficiency (Kbps/KB)} = \textit{Throughput [Kbps]} * (\textit{Code size [KB]})^{-1} \quad (2)$$

Power and Energy consumption: power is measured in micro Watt (µW) for hardware implementations and it is dependent on the clock frequency. Energy consumption per bit can be calculated as follows for both hardware and software implementations [63]:

$$\textit{Energy [µJ]} = (\textit{Latency [cycles/block]} \times \textit{Power [µW]})/\textit{block size [bits]} \quad (3)$$

Where,

Latency = the number of clock cycles required to encrypt one block of data,

Power = power consumed by the hardware or software implementation in µW,

Block size = size of data in bits can process in encryption/decryption operation.

Power can be Optimizing by minimizing the memory footprint of the source code and simplify the operations while maintaining a sufficient level of security.

6.2 Related works on Performance Evaluation of Lightweight Block Ciphers

Researches on performance evaluation take into accounts three directions: software, hardware and software / hardware evaluation papers. In this paper we focus on software related evaluation, we discuss different approaches or technologies for security and performance evaluation of lightweight ciphers based on restricted environment (platform), target applications and show their experimental Results as below:

- [49] 2022, Study the performance evaluation of two lightweight block cipher, AES and Saturnin to improve IoT applications Client-server model. They used ESP8266, the Node MCU 0.9 as restricted environment. Their Experimental Results shows that saturnin is twice faster than AES. The estimated Round Trip Time (RTT) to send and receive the data packet is being reduced to half.
- [50] 2021, propose CTR mode optimization technique by using parallel implementations of ARX-based block ciphers: LEA, HIGHT, and revised CHAM. They used Raspberry Pi 4B with ARM Cortex-A72 (64-bit processor) as restricted environment for IoT applications. Their Experimental Results shows an improvement the performance in LEA, HIGHT, and revised CHAM-64/128 ciphers.
- [51] 2021, compare the Evaluation metrics: RAM/ROM consumption, execution time, throughput, and energy consumption for ten lightweight block cipher: AES, PRESENT, LBlock, Skipjack, SIMON, XTEA, PRINCE, Piccolo, HIGHT and RECTANGLE by using Raspberry Pi 3(64-bit ARM Cortex processor) and Arduino Mega 2560 (ATmega2560 8-bit microcontroller) as restricted environment for IoT applications. Experimental Results show that the least amount of power is consumed by Skipjack, RECTANGLE, XTEA, and HIGHT and the highest measured power consumption is seen in Piccolo and PRESENT.
- [52] 2019, compare the Evaluation metrics: Throughput, Code Size, Used SRAM , Execution Time for a chosen lightweight block ciphers: AES, Roadrunner ,Simon ,Speck, Present ,Rectangle, Pride ,SparX, RC5, LED Lblock ,Fantomas Skinny. They used Arduino Uno (ATmega328 8-bit Microcontroller (MCU)) as restricted environment for IoT applications. Experimental Results shows that with respect of Code Size, Speck has the smallest code size (10% flash memory usage) and Fantomas has the largest code size (19% flash memory usage).For the Used SRAM, LED and Speck is the lest and AES, Present, Rectangle and Lblock most SRAM consuming. The worst execution time is Present and LED and the best is Speck. With respect to Throughput, Speck has the highest throughput value 28.58 bytes/ms and Present has the lowest throughput value with 0.06 bytes/ms.
- [53] 2020, presented a case study to secure communication between ultra-low-energy IoT devices. They used Nordic-Semiconductor nRF51822 ARM Cortex-M0 32 -bit processor as restricted environment to Benchmark the energy consumption (performance and memory consumption) of a large variety of crypto algorithms (block ciphers, stream ciphers, Authenticated Encryption with Associated Data (AEAD) and hash functions, Message Authentication Code (MAC) structures and digital signature) on a real MCU-based IoT device, and compare their results. They give over 170 encryption source code benchmarking reports results based on 450 experiments.
- [54] 2020, compare the throughput, energy, power consumption, RAM and ROM usage of several selected lightweight block ciphers (AES, CLEFIA, DES, Triple DES, TEA, XTEA, IDEA, PRESENT, SEA, SPECK, and TWOFISH) to find the most suitable cryptographic schema for IoT devices. Experimental results were obtained using Cooja simulator using z1 motes uses MSP430F2617 microcontroller (16-bit) architecture. SPECK and XTEA have outscored other algorithms in terms of throughput and energy usage.
- [55] 2018, compare the performance of several selected lightweight block ciphers (AES, SPECK, SIMON, Piccolo, HIGHT, PRESENT, LBlock, KLEIN) to evaluated the most suitable cryptographic schema for Industrial Wireless Sensor Networks by using

STM32F407ZGT6 microcontroller (32-bit ARM Cortex-M4 core). They are also examining the ciphers that possess good avalanche effect. By comparing different metrics, the code size, RAM size, Throughput, cycles/byte and the combined metric (Code-size× Cycle_count / Block_size), SPECK cipher shows good results.

- [56] 2018, the Performance Analysis of Different symmetric Cryptography Algorithms are compared, Stream Cipher: Rivest Cipher 4 (RC4) andChaCha20, and Block Ciphers: DES, 3DES, Blowfish, Twofish, Rivest Cipher 2 (RC2) and AES. Raspberry Pi 3 (64- bit ARM Cortex-A53 Processor) and Beagle Bone Black (32- bit ARM Cortex-A8 Processor) as restricted environment for IoT applications. Different data file sizes ranging from 1 MB - 128 MB), different key size and block size are compared to show the best execution time of the selected symmetric algorithms.
- [57] 2019, A Benchmark lightweight block ciphers framework was presented for embedded platforms. AVR ATmega128 (8-bit architecture), TI MSP430F1611 (6-bit platform) and ARM Cortex-M3 (32-bit RISC machine) are used to compare the Performance metrics (execution time, RAM consumption, and code size) of 19 lightweight block algorithm(AES, Chaskey, Fantomas, HIGHT, LEA, LED, LBlock, Piccolo, PRESENT, PRIDE, PRINCE, RoadRunneR, Robin, Simon, SPARX, RC5, RECTANGLE, Speck, and TWINE). Through results, ARX structure cipher is the belter regarding small RAM footprint and code size. Chaskey and Speck presented the best metrics results followed by Simon, LEA, RECTANGLE, and SPARX.
- [58] 2020, a comparison of the Performance metrics (RAM usage, CPU usage ,execution time and throughput) is done over various lightweight symmetric cipher (CLEFIA, Pride, Prince, KATAN, SKINNY, PRESENT, SPECK, SIMON, XTEA, AES-128, RC4, Rabbit, Trivium) and asymmetric ciphers(ELLI, RSA). By using MacBook Pro with Intel Core i7-5557Uand Raspberry Pi with 64-bit quad-core ARM Cortex-A72 processor as a testbed. Results show that Python implementation of the SPECK and SIMON ciphers are the most efficient with key size of 128 bits and The C implementation of CLEFIA-128 is more efficient rather than its python implementation

7. Design Strategies of Lightweight Block Cipher Algorithm

The optimal design of lightweight algorithms is based on a trade-off between cost, performance, and security requirements as shown in Figure 2 [62].

Cost vs. performance: Serial implementation is the minimal cost approach but degrades the performance with added loops, while the more number of simultaneous calculation and processing, the higher performance.

Performance vs. security: lower number of rounds possesses lower latency, while higher number of rounds is a safer cipher.

Security vs. cost: longer key length means more time requires attacking while, lower key length indicates less register and memory requirement.

The trade-off of these three criteria is subject to other conditions, including the application for which it is designed and the implementation environment based (H.W or S.W).

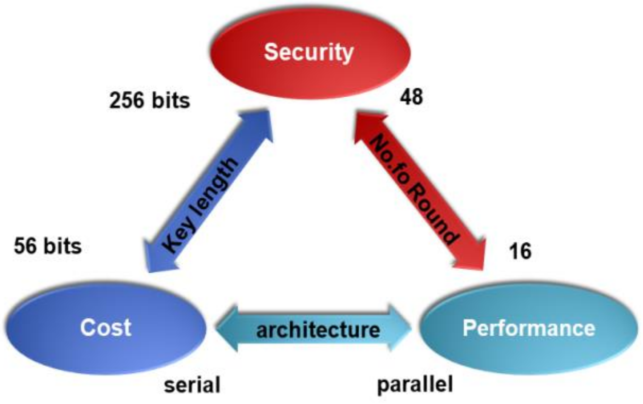

Fig. 2 Design strategies Trade-Offs in Lightweight Cryptography

Block cipher security depends on the confusion and diffusion principles, which Shannon had identified [60]. In a cipher, a nonlinear operation causes confusion and linear operations causes diffusion.

- Confusion
  Practically, confusion can be achieved by using costly S-boxes such as 8-bit S-box that used by the AES, a compact 4-bit S-boxes are used to low-cost hardware implementation. For lightweight designing ciphers, the look-up tables (LUTs) is alternative way for representing S-boxes which can improve throughput in software implementations through fast memory retrieval .A bitslice implementation by performing basic bitwise operations (XOR or AND) on words of w bits is another way [30].Although bitslice implementations can be very quick, they are limited to having a large memory overhead make it suitable for only non-feedback modes of operation like CTR mode [57]. ARX structures based ciphers can achieve low cost nonlinearity software implementation through modular addition operations, examples of this cipher Speck [26] and Sparx [39].
- Diffusion
  A good diffusion can be achieved by using bit permutation. For hardware implementation it is simply represented by bit-wise permutation such as diffusion layer that used by Present [47]. Bit rotation in word and

MDs matrices can achieve low cost permutation for software implementation [24].

- Simple key schedule to derive subkeys and Simple round function consists of simple operations.
- Block size (64 bit or less) and key length according to NIST, the smallest key size is 112 bits [6]. Devices' characteristics play a key role in determining the size of the block.
- Number of rounds: execution times will be lowered by reducing the number of rounds. The number of rounds inversely proportional to security level complexity of the confusion and diffusion layer, in [61] based on Speck cipher, a hybrid cipher Speck-R presented which reduced the number of rounds from 26 to 7 and the execution time at least 18% for Speck by integrating ARX structure with a dynamic substitution layer.
- Using bitwise operations and simple operations like modular addition can decrease the code size and RAM consumption.
- The word size used in cipher operation should be on par with the largest register size that is supported by constricted architectures.

8. Conclusion

Researchers work to improve security levels and strengthen ciphers against both existing and new threats. This paper discusses the most prominent security problems of restricted devices in the IoT environment.

Encryption is one of the most effective methods for providing end- to- end security. Lightweight cryptographic algorithm is essential for handling security in highly constrained environments such as the Internet of Things. Block cipher is very convenient and easier to implement in software, it can be operated on data in computer-sized blocks. Due to many considerations such as energy and memory utilization, especially for software platforms, this article will assist IoT security developers to highlight algorithms that match the needs of the constrained environment. Through the design strategies presented in this research, it can be said that designing a cipher with simple round functions and simple operations can achieve a high or acceptable level of security but it depends on the requirements and specifications of the target application and the constraints of the device. It should be emphasized that it is necessary to evaluate the performance on different resource constrained devices to develop a more comprehensive understanding of the lightweight ciphers.

References

[1] Meng, T. X. and Buchanan, W. (2020). Lightweight cryptographic algorithms on resource-constrained devices. Preprints.

[2] Eisenbarth, T., Kumar, S., Paar, C., Poschmann, A., and Uhsadel, L. (2007). A survey of lightweight-cryptography implementations. IEEE Design & Test of Computers, 24(6), 522-533.

[3] Hassan, W. H. (2019). Current research on Internet of Things (IoT) security: A survey. Computer networks, 148, 283-294.

[4] Biryukov, A., and Perrin, L. (2017). State of the art in lightweight symmetric cryptography. Cryptology ePrint Archive.

[5] Serpanos, D. N., and Voyiatzis, A. G. (2013). Security challenges in embedded systems. ACM Transactions on embedded computing systems (TECS), 12(1s), 1-10.

[6] McKay, K., Bassham, L., Sönmez Turan, M., and Mouha, N. (2016). Report on lightweight cryptography (No. NIST Internal or Interagency Report (NISTIR) 8114 (Draft)). National Institute of Standards and Technology.

[7] Poojari, A., and Nagesh, H. R. (2019). A comparative analysis of symmetric lightweight block ciphers. In Emerging Technologies in Data Mining and Information Security (pp. 705-711). Springer, Singapore.

[8] Thakor, V. A., Razzaque, M. A., and Khandaker, M. R. (2021). Lightweight cryptography algorithms for resource-constrained IoT devices: A review, comparison and research opportunities. IEEE Access, 9, 28177-28193.

[9] Poschmann, A., Leander, G., Schramm, K., and Paar, C. (2007, May). New light-weight crypto algorithms for RFID. In 2007 IEEE International Symposium on Circuits and Systems (pp. 1843-1846). IEEE.

[10] Dinu, D., Biryukov, A., Großschädl, J., Khovratovich, D., Le Corre, Y., and Perrin, L. (2015, July). FELICS–fair evaluation of lightweight cryptographic systems. In NIST Workshop on Lightweight Cryptography (Vol. 128).

[11] Wheeler, D. J., and Needham, R. M. (1994, December). TEA, a tiny encryption algorithm. In International workshop on fast software encryption (pp. 363-366). Springer, Berlin, Heidelberg.

[12] Needham, R. M., and Wheeler, D. J. (1997). Tea extensions. Report (Cambridge University, Cambridge, UK, 1997).

[13] Matsui, M. (1997, January). New block encryption algorithm MISTY. In International Workshop on Fast Software Encryption (pp. 54-68). Springer, Berlin, Heidelberg.

[14] Kitsos, P., Galanis, M. D., and Koufopavlou, O. (2004, May). High-speed hardware implementations of the KASUMI block cipher. In 2004 IEEE International Symposium on Circuits and Systems (IEEE Cat. No. 04CH37512) (Vol. 2, pp. II-549). IEEE.

[15] Daemen, J., and Rijmen, V. (1999). AES proposal: Rijndael.

[16] Aoki, K., Ichikawa, T., Kanda, M., Matsui, M., Moriai, S., Nakajima, J., and Tokita, T. (2000, August). Camellia: A 128-bit block cipher suitable for multiple platforms—design and analysis. In International workshop on selected areas in cryptography (pp. 39-56). Springer, Berlin, Heidelberg.

[17] Hong, D., Sung, J., Hong, S., Lim, J., Lee, S., Koo, B. S., and Chee, S. (2006, October). HIGHT: A new block cipher suitable for low-resource device. In International workshop on cryptographic hardware and embedded systems (pp. 46-59). Springer, Berlin, Heidelberg.

[18] Kim, B., Cho, J., Choi, B., Park, J., and Seo, H. (2019). Compact implementations of HIGHT block cipher on IoT platforms. Security and Communication Networks, 2019.

[19] Standaert, F. X., Piret, G., Gershenfeld, N., and Quisquater, J. J. (2006, April). SEA: A scalable encryption algorithm for small embedded applications. In International Conference on Smart Card Research and Advanced Applications (pp. 222-236). Springer, Berlin, Heidelberg.

[20] Shirai, T., Shibutani, K., Akishita, T., Moriai, S., and Iwata, T. (2007, March). The 128-bit blockcipher CLEFIA. In International workshop on fast software encryption (pp. 181-195). Springer, Berlin, Heidelberg.

[21] Gong, Z.,Nikova, S., and Law, Y. W. (2011, June). KLEIN: a new family of lightweight block ciphers. In International workshop on radio frequency identification: security and privacy issues (pp. 1-18). Springer, Berlin, Heidelberg.

[22] Ning, L., Ali, Y., Ke, H., Nazir, S., and Huanli, Z. (2020). A hybrid MCDM approach of selecting lightweight cryptographic cipher based on ISO and NIST lightweight cryptography security requirements for internet of health things. IEEE Access, 8, 220165-220187.

[23] Wu, W., and Zhang, L. (2011, June). LBlock: a lightweight block cipher. In International conference on applied cryptography and network security (pp. 327-344). Springer, Berlin, Heidelberg.

[24] Guo, J., Peyrin, T., Poschmann, A., and Robshaw, M. (2011, September). The LED block cipher. In International workshop on cryptographic hardware and embedded systems (pp. 326-341). Springer, Berlin, Heidelberg.

[25] Suzaki, T., Minematsu, K., Morioka, S., and Kobayashi, E. (2011, November). Twine: A lightweight, versatile block cipher. In ECRYPT workshop on lightweight cryptography (Vol. 2011).

[26] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L. (2013). The SIMON and SPECK families of lightweight block ciphers. cryptology eprint archive.

[27] Beaulieu, R., Shors, D., Smith, J., Treatman-Clark, S., Weeks, B., and Wingers, L. (2015). SIMON and SPECK: Block Ciphers for the Internet of Things. Cryptology ePrint Archive.

[28] Karakoç, F., Demirci, H., and Harmancı, A. E. (2013, May). ITUbee: a software oriented lightweight block cipher. In International Workshop on Lightweight Cryptography for Security and Privacy (pp. 16-27). Springer, Berlin, Heidelberg.

[29] Mouha, N., Mennink, B., Herrewege, A. V., Watanabe, D., Preneel, B., and Verbauwhede, I. (2014, August). Chaskey: an efficient MAC algorithm for 32-bit microcontrollers. In International conference on selected areas in cryptography (pp. 306-323). Springer, Cham.

[30] Grosso, V., Leurent, G., Standaert, F. X., and Varıcı, K. (2014, March). LS-designs: Bitslice encryption for efficient masked software implementations. In International Workshop on fast software encryption (pp. 18-37). Springer, Berlin, Heidelberg.

[31] Journault, A., Standaert, F. X., and Varici, K. (2017). Improving the security and efficiency of block ciphers based on LS-designs. Designs, Codes and Cryptography, 82(1), 495-509.

[32] Kumar, M., Sk, P. A. L., and Panigrahi, A. (2014). FeW: a lightweight block cipher. Turkish Journal of Mathematics and Computer Science, 11(2), 58-73.

[33] Albrecht, M. R.,Driessen, B., Kavun, E. B., Leander, G., Paar, C., and Yalçın, T. (2014, August). Block ciphers–focus on the linear layer (feat. PRIDE). In Annual Cryptology Conference (pp. 57-76). Springer, Berlin, Heidelberg.

[34] Zhang, W., Bao, Z., Lin, D., Rijmen, V., Yang, B., and Verbauwhede, I. (2015). RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms. Science China Information Sciences, 58(12), 1-15.

[35] Yang, G., Zhu, B., Suder, V., Aagaard, M. D., and Gong, G. (2015, September). The simeck family of lightweight block ciphers. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 307-329). Springer, Berlin, Heidelberg.

[36] Encarnacion, P. C., Gerardo, B. D., and Hernandez, A. A. (2020, June). Performance Analysis on Enhanced Round Function of SIMECK Block Cipher. In 2020 12th International Conference on Communication Software and Networks (ICCSN) (pp. 270-275). IEEE.

[37] Park, T., Seo, H., Lee, G., and Kim, H. (2017, July). Efficient implementation of simeck family blocks cipher on 16-bit MSP430. In 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN) (pp. 983-988). IEEE.

[38] Baysal, A., and Şahin, S. (2015, September). Roadrunner: A small and fast bitslice block cipher for low cost 8-bit processors. In Lightweight Cryptography for Security and Privacy (pp. 58-76). Springer, Cham.

[39] Dinu, D., Perrin, L., Udovenko, A., Velichkov, V., Großschädl, J., and Biryukov, A. (2016). Sparx: a family of ARX-based lightweight block ciphers provably secure against linear and differential attacks. In NIST Lightweight Cryptography Workshop 2016.

[40] Dinu, D., Perrin, L., Udovenko, A., Velichkov, V., Großschädl, J., and Biryukov, A. (2016, December). Design strategies for ARX with provable bounds: Sparx and LAX. In International Conference on the Theory and Application of Cryptology and Information Security (pp. 484-513). Springer, Berlin, Heidelberg.

[41] Bansod, G., Patil, A., Sutar, S., and Pisharoty, N. (2016). ANU: an ultra lightweight cipher design for security in IoT. Security and Communication Networks, 9(18), 5238-5251.

[42] Bansod, G., Pisharoty, N., and Patil, A. (2016). PICO: An Ultra Lightweight and Low Power Encryption Design for Ubiquitous Computing. Defence Science Journal, 66(3).

[43] Beierle, C., Jean, J., Kölbl, S., Leander, G., Moradi, A., Peyrin, T., ... and Sim, S. M. (2016, August). The SKINNY family of block ciphers and its low-latency variant MANTIS. In Annual International Cryptology Conference (pp. 123-153). Springer, Berlin, Heidelberg.

[44] Usman, M., Ahmed, I., Aslam, M. I., Khan, S., and Shah, U. A. (2017). SIT: a lightweight encryption algorithm for secure internet of things. arXiv preprint arXiv:1704.08688.

[45] Xia, X., Chen, B., and Zhong, W. (2021, July). Correlation Power Analysis of Lightweight Block Cipher Algorithm LiCi. In Journal of Physics: Conference Series (Vol. 1972, No. 1, p. 012055). IOP Publishing.

[46] Koo, B., Roh, D., Kim, H., Jung, Y., Lee, D. G., and Kwon, D. (2017, November). CHAM: A family of lightweight block ciphers for resource-constrained devices. In International conference on information security and cryptology (pp. 3-25). Springer, Cham.

[47] Banik, S., Pandey, S. K., Peyrin, T., Sasaki, Y., Sim, S. M., and Todo, Y. (2017, September). GIFT: a small present. In International conference on cryptographic hardware and embedded systems (pp. 321-345). Springer, Cham.

[48] Sehrawat, D., and Gill, N. S. (2019). Performance evaluation of newly proposed lightweight cipher, BRIGHT. Int. J. Intell. Eng. Syst, 12(4), 71-80.

[49] Podimatas, P., and Limniotis, K. (2022). Evaluating the Performance of Lightweight Ciphers in Constrained Environments—The Case of Saturnin. Signals, 3(1), 86-94.

[50] Song, J., and Seo, S. C. (2021). Efficient parallel implementation of CTR mode of ARX-based block ciphers on ARMv8 microcontrollers. Applied Sciences, 11(6), 2548.

[51] Panahi, P., Bayılmış, C., Çavuşoğlu, U., and Kaçar, S. (2021). Performance evaluation of lightweight encryption algorithms for IoT-based applications. Arabian Journal for Science and Engineering, 46(4), 4015-4037.

[52] Polat, S. (2019). Performance evaluation of lightweight cryptographic algorithms for internet of things security (Master's thesis, Middle East Technical University).

[53] Aerabi, E., Bohlouli, M., Livany, M. H. A., Fazeli, M., Papadimitriou, A., and Hely, D. (2020). Design space exploration for ultra-low-energy and secure IoT MCUs. ACM Transactions on Embedded Computing Systems (TECS), 19(3), 1-34.

[54] Makarenko, I., Semushin, S., Suhai, S., Kazmi, S. A., Oracevic, A., and Hussain, R. (2020, October). A comparative analysis of cryptographic algorithms in the internet of things. In 2020 International Scientific and Technical Conference Modern Computer Network Technologies (MoNeTeC) (pp. 1-8). IEEE.

[55] Pei, C., Xiao, Y., Liang, W., and Han, X. (2018). Trade-off of security and performance of lightweight block ciphers in Industrial Wireless Sensor Networks. EURASIP Journal on Wireless Communications and Networking, 2018(1), 1-18.

[56] Singh, P., and Deshpande, K. (2018). Performance evaluation of cryptographic ciphers on IoT devices. arXiv preprint arXiv:1812.02220.

[57] Dinu, D., Corre, Y. L., Khovratovich, D., Perrin, L., Großschädl, J., and Biryukov, A. (2019). Triathlon of lightweight block ciphers for the internet of things. Journal of Cryptographic Engineering, 9(3), 283-302.

[58] Meng, T. X., and Buchanan, W. (2020). Lightweight cryptographic algorithms on resource-constrained devices. Preprints.

[59] Thabit, F., Alhomdy, S., Al-Ahdal, A. H., and Jagtap, S. (2021). A new lightweight cryptographic algorithm for enhancing data security in cloud computing. Global Transitions Proceedings, 2(1), 91-99.

[60] Shannon, C. E. (1949). Communication theory of secrecy systems. The Bell system technical journal, 28(4), 656-715.

[61] Sleem, L., and Couturier, R. (2021). Speck-R: An ultra light-weight cryptographic scheme for Internet of Things. Multimedia Tools and Applications, 80(11), 17067-17102.

[62] Poschmann, A. Y. (2009). Lightweight cryptography: cryptographic engineering for a pervasive world. In Ph. D. Thesis.

[63] Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I., and Manifavas, C. (2018). A review of lightweight block ciphers. Journal of cryptographic Engineering, 8(2), 141-184.

[64] Rao, P. M., and Deebak, B. D. (2022). Security and privacy issues in smart cities/industries: technologies, applications, and challenges. Journal of Ambient Intelligence and Humanized Computing, 1-37.

[65] Fan, J., Yang, W., and Lam, K. Y. (2022). Cybersecurity Challenges Of IoT-enabled Smart Cities: A Survey. arXiv preprint arXiv:2202.05023.

حلول تنفيذ برمجيات كتل التشفير خفيفة الوزن لتأمين بيئة انترنت الأشياء المقيدة

رؤى مؤيد الياس / قسم علوم الحاسوب , كلية علوم الحاسوب والرياضيات,
جامعة الموصل , الموصل, العراق.
ruaa.moayad@uomosul.edu.iq

سفيان سالم محمود الدباغ / قسم علوم الحاسوب , كلية علوم الحاسوب
والرياضيات, جامعة الموصل ,الموصل, العراق.
drsufyan.salim@uomosul.edu.iq

تاريخ الاستلام:2022 /30/6 تاريخ القبول:10/10/2022

الملخص

مع تطور تقنية إنترنت الأشياء (IoT) تم دمج أجهزة إنترنت الأشياء في العديد من التطبيقات التي تخص حياتنا اليومية ، بما في ذلك التطبيقات الصناعية والأمنية والطبية والشخصية. ظهرت العديد من انتهاكات أمان إنترنت الأشياء بسبب البنية التحتية المادية الحرجة ، ونقاط ضعف الشبكة. بالنظر إلى طبيعة الموارد المقيدة والمحدودة لهذه الأجهزة من حيث الحجم والسعة والطاقة ، أصبح الأمن مهمًا بشكل متزايد. يعد التشفير الخفيف أحد الاتجاهات التي تقدم حلولاً أمنية في البيئات محدودة الموارد مثل تحديد الهوية باستخدام الترددات الراديوية (RFID) وشبكة المستشعرات اللاسلكية . (WSN) تناقش هذه الورقة مشكلات الأمان الخاصة بأجهزة إنترنت الأشياء المحدودة الموارد وتستعرض أبرز تشفير كتلي خفيف الوزن مناسب لتنفيذ البرامج. من خلال دراسة المواصفات والهيكل الداخلي لكل تشفير وتقييم أداء تنفيذها على نوع محدد من المنصات ، نقدم دليلًا إرشاديًا لأستراتيجيات التصميم لمطوري التشفير لتصميم حلول محسّنة لتشفير كتلي خفيفة الوزن وتنفيذ برمجي مدمج للبيئات محدودة الموارد.

الكلمات المفتاحية: التشفيرالكتلي ،أمن أجهزة إنترنت الأشياء ، التشفير خفيف الوزن ، تقييم الأداء ، الأجهزة محدودة الموارد.