IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received December 9, 2020, accepted January 9, 2021, date of publication January 19, 2021, date of current version February 22, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3052867

Lightweight Cryptography Algorithms for Resource-Constrained IoT Devices: A Review, Comparison and Research Opportunities

VISHAL A. THAKOR[1], **MOHAMMAD ABDUR RAZZAQUE**[1], **(Member, IEEE), AND MUHAMMAD R. A. KHANDAKER**[2], **(Senior Member, IEEE)**
[1]School of Computing, Engineering, and Digital Technologies, Teesside University, Middlesbrough TS1 3BX, U.K.
[2]School of Engineering and Physical Sciences, Heriot-Watt University, Edinburgh EH14 4AS, U.K.

Corresponding author: Mohammad Abdur Razzaque (m.razzaque@tees.ac.uk)

This research was partially funded by the Newton Fund - Institutional Links from British Council and National Research Council of Thailand with the grant ID is 527643161.

ABSTRACT IoT is becoming more common and popular due to its wide range of applications in various domains. They collect data from the real environment and transfer it over the networks. There are many challenges while deploying IoT in a real-world, varying from tiny sensors to servers. Security is considered as the number one challenge in IoT deployments, as most of the IoT devices are physically accessible in the real world and many of them are limited in resources (such as energy, memory, processing power and even physical space). In this paper, we are focusing on these resource-constrained IoT devices (such as RFID tags, sensors, smart cards, etc.) as securing them in such circumstances is a challenging task. The communication from such devices can be secured by a mean of lightweight cryptography, a lighter version of cryptography. More than fifty lightweight cryptography (plain encryption) algorithms are available in the market with a focus on a specific application(s), and another 57 algorithms have been submitted by the researchers to the NIST competition recently. To provide a holistic view of the area, in this paper, we have compared the existing algorithms in terms of implementation cost, hardware and software performances and attack resistance properties. Also, we have discussed the demand and a direction for new research in the area of lightweight cryptography to optimize balance amongst cost, performance and security.

INDEX TERMS IoT, lightweight, cryptography, sensors, RFID, smart cards.

I. INTRODUCTION

A. IoT OVERVIEW

Internet of Things (IoT) has already become a dominant research era because of its applications in various domains such as smart transport & logistics, smart healthcare, smart environment, smart infrastructure (smart cities, smart homes, smart offices, smart malls, Industry 4.0), smart agriculture and many more. Many researchers and industry experts have given various definitions of IoT depending on their applications and implementation area, but in simple words, IoT is a network of connected things, each with a unique identification, able to collect and exchange data over the Internet with or without human interaction [1]–[5]. In any IoT solution or application, IoT devices are the key elements. These IoT devices could be divided into two main categories (Figure 1): rich in resources such as servers, personal computers, tablets and smartphones, etc. and limited in resources (resource-constrained) such as industrial sensors or sensor nodes, RFID tags, actuators, etc., [6]. In this paper, we focus on the second category of IoT devices. These connected devices are becoming more popular due to their use in various application and will flood the market with the emergence of IoT [6], leading an enormous data exchange rate amongst [7].

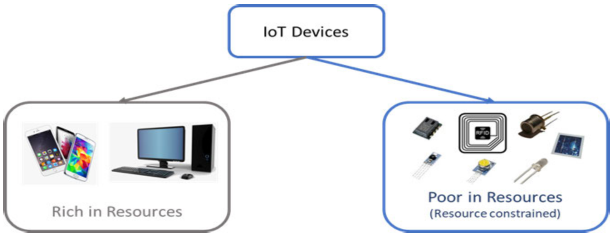


FIGURE 1. Two main categories of IoT Devices.

The associate editor coordinating the review of this manuscript and approving it for publication was Kim-Kwang Raymond Choo.

TABLE 1. List of Abbreviations and Acronyms.

Acronyms	Abbreviations
RFID	Radio Frequency IDentification
LWC	Lightweight Cryptography
FPGA	Field Programmable Gate Arrays
SPN	Substitution-Permutation Network
FN	Feistel Network
GFN	General Feistel Network
ARX	Add-Rotate-XOR
NLFSR	NonLinear-Feedback Shift Register
AEAD	Authenticated Encryption with Associated Data
GSM	Global System for Mobile Communications
UMTS	Universal Mobile Telecommunications System
GPRS	General Packet Radio Service
NIST	National Institute of Standards and Technology (USA)
ISO/IEC	International Organization of Standardization and the International Electrotechnical Commission
IETF	Internet Engineering Task Force
NESSIE	New European Schemes for Signatures, Integrity and Encryption
Cryptrec	Cryptography Research and Evaluation Committees (Japan)
Ecrypt	European Network of Excellence in Cryptology
NSA	National Security Agency (USA)
MITM	Man-in-the-Middle Attack
GE	Gate Equivalents
S-box	Substitution box
LED	Lightweight Encryption Device
EPCBC	Electronic Product Code Block Cipher
DES	Data Encryption Standard
AES	Advanced data Encryption Standard
TEA	Tiny Encryption Algorithm
XTEA	eXtended TEA
SEA	Scalable Encryption Algorithm
IDEA	International Data Encryption Algorithm
HIGHT	High security and lightweight
BEST-1	Better Encryption Security Technique-1
LEA	Lightweight block Encryption Algorithm
ETRIK	Electronics and Telecommunication Research Institute of Korea

B. SECURITY CONCERNS OF RESOURCE-CONSTRAINED IoT DEVICES: CHALLENGES AND SECURITY REQUIREMENTS

When billions of smart devices (connected devices) working in a diverse set of platforms, especially when shifting from server to sensors, gives birth to various unprecedented challenges to their owners or users [6] such as security & privacy, interoperability, longevity & support, technologies and many more [8]. Also, IoT devices are easily accessible and exposed to many security attacks [9] as they interact directly with the physical world to collect confidential data or to control physical environment variables, which makes them an attractive target for attackers [10]. All these circumstances make cybersecurity as a major challenge in IoT devices with demands of confidentiality, data integrity, authentication & authorization, availability, privacy & regulation standards and regular system updates [8]. The Figure 2 depicts IoT security challenges and its security requirements.

In this scenario, cryptography could be one of the effective measures to guarantee confidentiality, integrity and authentication & authorization of the traversing data through IoT devices [7]. It could also be a solution to secure the stored or traversing data over the network. However, conventional PC based cryptography algorithms do not fit into resource-constrained IoT devices due to their high resource demands. A lighter version of these solutions, lightweight cryptography, can address these challenges to secure the communication in resource-constrained IoT devices.

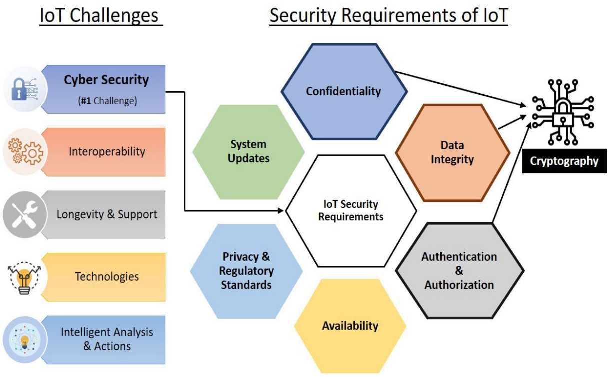


FIGURE 2. IoT Security Challenges.

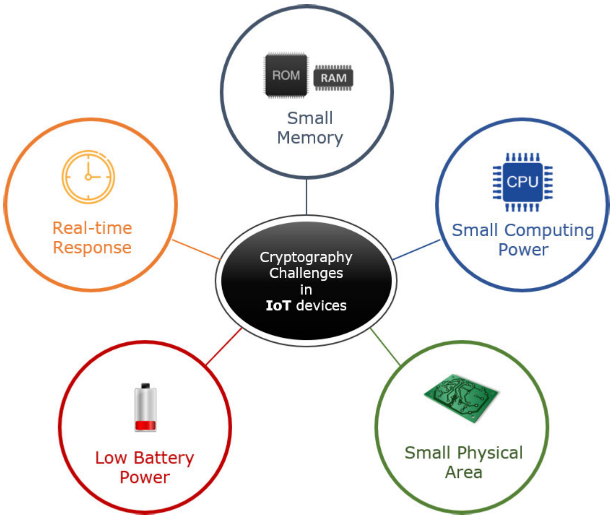


FIGURE 3. Key Challenges with Conventional Cryptography.

C. KEY CHALLENGES WHILE IMPLEMENTING CONVENTIONAL CRYPTOGRAPHY IN RESOURCE-CONSTRAINED IoT DEVICES

The key challenges while implementing conventional cryptography in IoT devices (Figure 3) are as follows [11]:

- Limited memory (registers, RAM, ROM)
- Reduced computing power
- Small physical area to implement the assembly
- Low battery power (or no battery)
- Real-time response

Most of the IoT devices (such as RFIDs and sensors) are small in size and are equipped with limited resources such as small memory (RAM, ROM) to store and to run the application, low computing power to process the data, limited battery power (or no battery in case of passive RFID tags) [6], small physical area to fit-in the assembly [6], [11]. Moreover, most of the IoT devices deal with the real-time application where quick and accurate response with essential security using available resources is a challenging task [12], [13]. IoT device designers face several risks and challenges, including energy capacity [14], and data security [9].

In these circumstances, if conventional cryptography standards are applied to IoT devices (mainly RFIDs and sensors), their performance may not be acceptable [6]. The above issues with conventional cryptography are very well addressed by its sub-discipline, lightweight cryptography, by introducing lightweight features such as small memory, small processing power, low power consumption, real-time response even with resource-constrained devices [6].

Another important aspect of lightweight cryptography is that it is not just applicable to resource-constrained devices (RFID tags, sensors, etc.), but readily applicable to other devices rich in resources that it directly or indirectly interacts with (such as servers, PCs, tablets, smartphones, etc.) [6].

D. MOTIVATION AND CONTRIBUTION

Recently, many algorithms have been proposed for LWC by the researchers. Besides, many works have revealed the security attacks on particular LWC algorithm(s) [15]–[31]. A number of published papers have done a fair comparison of hardware and/or software implementations of these algorithms on different platforms as well as in different circumstances [9], [32]–[39]. Most of these works have considered the algorithms which are applicable in certain domains or suitable for certain applications. However, a holistic view of the proposed LWC algorithms in terms of their hardware-software performances along with cryptanalysis is missing in these works. Authors in [40] have reviewed a list of different LWC algorithms with their performances on different platforms but missing an inclusive view on their applications and lightweight key demands of cost (memory, physical area, battery, power) and performance (quick response) along with the security concerns. Also, [40] does not include a number of key algorithms, e.g., Keeloq and Midori. In addition, it just provides a list of attacks on LWC algorithms without any security comparison, and thus a clear view of various security attacks on different LWC algorithms is missing.

More recently, [41] discusses on the algorithms, especially submitted to the NIST competition (round 2), which are compliant with LWC Hardware API (proposed by the NIST in 2019) and evaluates them on FPGA platform (Xilinx, Intel, and Lattice). The paper considers only two performance metrics: Throughput and Speed (clock-cycles/byte) which could be its limitation as others (Block/Key size, Memory, Gate Area, Power & Energy requirements) are missing. Also, these algorithms are running in a competition through several rounds (32 out of 57 (in round 1) are competing in the 2nd round).

With a unique aspect in this paper, we have clearly classified the key characteristics of LWC algorithms (missing in the existing survey papers) proposed by the leading research groups [6], [42] in the fields of cryptography along with how LWC satisfies these properties (Table 2). Secondly, our paper compares 41 existing symmetric key lightweight cryptography (plain encryption) algorithms over 7 performance metrics (Block/Key size, Memory, Gate Area, Latency, Throughput, Power & Energy requirements along with hardware and software efficiency) as recommended by the NIST report for resource-constrained IoT devices [6]. These LWC algorithms are widely adopted by the industries and the article reveals the top ten amongst them based on their mapping (metrics). These analyses could be useful to researchers/scientists in choosing the right algorithm based on their application requirement(s). Also, demonstrating various IoT applications in real-world along with their lightweight key requirements and their best suite LWC options is a unique contribution in the field of lightweight cryptography. In addition, our paper evaluates various attacks on different LWC algorithms in a grid form. Such comparison eases users to identify the security strength of any LWC algorithm as well as to identify common attacks on LWC algorithms. A recent call from NIST [43] (to create new LWC algorithms for easy and efficient implementation on resource-constrained circuitry) and the results derived from the study (none of the algorithms meets all the criteria of lightweight in terms of cost and performance along with strong security), really encourage to explore the existing list of LWC algorithms from different perspectives for further research.

TABLE 2. LWC Characteristics.

Characteristics		What LWC can offer?
Physical (Cost)	Physical Area (GEs, logic blocks)	-Tiny key & block -Simple rounds with simple computation -Simple key generation
	Memory (registers, RAM, ROM)	
	Battery power (energy consumption)	
Performance	Computing power (latency, throughput)	
Security	Minimum security strength (bits)	-Strong internal structure
	Attack models (related key, multi-keys)	
	Side-channel and Fault-injection attacks	

E. PAPER OUTLINE

Considering the significance of IoT security, this article takes an inclusive view on symmetric key lightweight cryptography algorithms and i) defines hardware and software performance metrics based on identified key characteristics of LWC and gives a broad classification of LWC based on their internal structure (Section II), ii) a comprehensive study of existing LWC algorithms along with their performances, cryptanalysis and real-time use cases (Section III), iii) outlines open research challenges, recommending future research directions (Section IV), and finally iv) concludes in (Section V).

II. LIGHTWEIGHT CRYPTOGRAPHY FOR RESOURCE-CONSTRAINED IoT DEVICES

A. CHARACTERISTICS OFFERED BY LWC

The three main characteristics of Lightweight cryptography algorithms and their offerings are listed in Table 2 [9], [11]:

As shown in the above table, physical cost, performance and security are the main characteristics to look into while implementing cryptography to any resource-constrained IoT device. Each of these characteristics is further observed where physical space occupied, memory demand and energy consumption as a cost to implement, processing power in terms of latency and through as performance (speed) and block/key length and different attack models including side-channel & fault-injection attacks as a security measure. First two characteristics are satisfied by LWC algorithms by offering simple round functions on the tiny block ($\leq$ 64bit) using a tiny key ($\leq$ 80bit) with simple key scheduling. The last but important characteristic, security, is fulfilled by the adoption of one of the six internal structures (SPN, FN, GFN, ARX, NLFSR, Hybrid) to immune against the security attacks.

B. HARDWARE AND SOFTWARE PERFORMANCE METRICS

Based on first two characteristics (physical and performance) offered by any LWC algorithms, hardware and software specific resource requirement could be measured in terms of memory requirements, gate area, latency, throughput, and power and energy consumption as follows:

1) MEMORY REQUIREMENTS

Generally, measured in *KB* [40]. RAM is required to store intermediate values that can be used in computations and ROM is required to store the program/algorithm, and static data, such as algorithm key, S-box (in some cases), etc., [6].

2) GATE AREA

It is the physical area required to implement/run the algorithm on a board/circuit, measured in μm^2. This space can be specified using logical blocks for FPGA or using GE for ASIC (1GE = 2 input-NAND Gate) [6]. Normally, 200 to 2000 GE (out of 1000 to 10,000 GE of total available) are allocated for security reasons in an economical RFID tag [44].

3) LATENCY

It is the time to produce the cipher from the original text in terms of hardware performance [6] whereas the amount of clock cycles per block (during encryption) defines the software latency.

4) THROUGHPUT

Throughput, in hardware, can be measured in terms of plain text processed per time unit (bits per second) at 100 *KHz* frequency, whereas in software, it is the average amount of plaintext processed per CPU clock cycle at 4 *MHz* frequency [45].

5) POWER REQUIREMENTS

The amount of power required by the circuit to process the algorithm can be measured in μW.

6) ENERGY CONSUMPTION

Energy consumption per bit can be calculated as follows [40]:

$$Energy[\mu J] = (Latency[cycles/block] * Power[\mu W]) \\ /blocksize[bits]$$

Here, latency is in terms of software implementation.

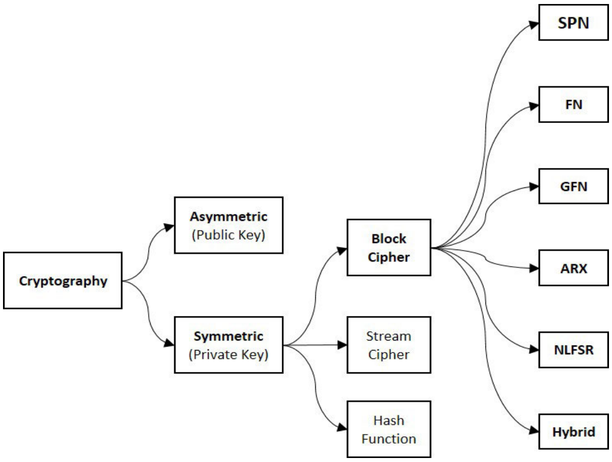


FIGURE 4. Structure wise Classification of LWC.

7) EFFICIENCY

Gives performance over resource requirements. For hardware, it can be calculated as follows [40]:

$$Hardware\ Efficiency \\ = Throughput[Kbps]/Complexity[KGE]$$

Here, complexity means physical space.

Similarly, **software efficiency** can be determined as follows [40]:

$$Software\ Efficiency = Throughput[Kbps]/CodeSize[KB]$$

Here, code size is the algorithm size.

C. STRUCTURE WISE CLASSIFICATION OF LWC

Cryptographic algorithms can be classified into two main categories, symmetric key and asymmetric key (Figure 4) cipher. Symmetric key uses a single key for both encryption and decryption of the data, whereas asymmetric cipher uses two different keys to encrypt and to decrypt the data [46]. Symmetric key cryptography is safe and comparatively fast, the only downside of symmetric key encryption is the sharing of key between the communicating parties without compromising it [32]. But this could be overcome by pre-sharing the key through a trusted third party. Also, it ensures confidentiality, data integrity and authentication (using authentication encryption mode (AEAD)) of the data. Asymmetric cryptography uses two private-public key pairs. It ensures confidentiality and integrity by making use of the public key of the receiver and further ensures authentication by using the sender's private key (as a digital signature) to encrypt the data. At the other end, the receiver decrypts it by using the sender's public key first and then using his/her private key [46]. The only disadvantage of asymmetric encryption is its large key which increases the complexity and slows down the process [32].

In block cipher, both encryption and decryption take place on a fixed size block (64 bits or more) at a time whereas stream cipher continuously processes the input elements bit by bit (or word by word) [46]. There are two fundamental properties of any cryptography, confusion and diffusion, introduced by Claude Shannon [35], [40] to strengthen the cipher. The confusion makes the relationship between the ciphertext and the key as complex as possible using substitution (S-box) whereas diffusion dissipates the statistical structure of plaintext over the bulk of ciphertext using permutation [35], [46]. The stream cipher uses only confusion property whereas block cipher uses both confusion and diffusion with simple design compared to the stream one. Following the reverse of encryption process to extract the original text is hard in a block cipher whereas stream cipher performs XOR function(s) to encrypt the data that could be easily reverted to its original form. In contrary, Hash is a one-way mathematical function that transforms unspecified length data into a specified-length bit string (short string) which cannot be inverted.

For the above reasons, a block cipher is preferred in resource-constrained IoT devices over stream cipher. This paper concentrates on block cipher, mainly symmetric lightweight block ciphers. It uses one of the following structure:

- Substitution-Permutation Network (SPN)
- Feistel Network (FN)
- General Feistel Network (GFN)
- Add-Rotate-XOR (ARX)
- NonLinear-Feedback Shift Register (NLFSR)
- Hybrid

Substitution-Permutation network (SPN) tweaks the data through a set of substitution box and permutation table and formulates them for the following round. A **Feistel network (FN)** breaks the input block into equal halves and applies diffusion in each round to just one half. In addition, swapping of two halves happens at the beginning of each round. The **generalized Feistel network (GFN)** is an extrapolated version of the classic Feistel network. It splits the input block into a number of sub-blocks and applies the Feistel functions to every pair of sub-blocks, followed by a cyclic shift proportional to the number of sub-blocks [47]. **ARX** performs encryption-decryption using addition, rotation and XOR functions without making use of S-box. Implementation of ARX is fast and compact but limits in security properties compared to SPN and Feistel ciphers. **Nonlinear feedback shift register (NLFSR)**, applies to both stream and block ciphers, utilizes the building blocks of stream ciphers whose current state is derived from its prior state which is a nonlinear feedback value [20]. **Hybrid** cipher combines any three types (SPN, FN, GFN, ARX, NLFSR) or even mixes block and stream property to improve specific characteristics (for example, throughput, energy, GE, etc.) based on its application requirements.

Out of these structures, SPN and FN are the most popular choice due to their flexibility to implement, based on application requirements [40]. Although Feistel structures are incorporated easily into low-average power hardware (due to the absence of round function in one-half of the states), it usually requires more round function compared to SPN structures for safety reasons [48]. When there is a choice between fewer SPN function rounds and higher Feistel function rounds with the same level of security and similar energy costs, SPN function could be a smarter choice [48].

III. EXISTING LWC ALGORITHMS

More than fifty symmetric LWC algorithms (plain encryption) are proposed by various academia, proprietaries and government bodies with a focus on reducing cost (memory, processing power, physical area (GE), energy consumption) and enhanced hardware and software performance (latency, throughput). However, many of them do not concentrate on security attacks explicitly and only care about performance and/or implementation cost [13]. The structure-wise categorisation of these algorithms is summarised in Table 3. The following subsections unfold these LWC algorithms category wise.

TABLE 3. Structure wise LWC algorithms.

Structure Type	Algorithms
SPN	AES, Present, GIFT, SKINNY, Rectangle, Midori, mCrypton, Noekeon, Iceberg, Puffin-2, Prince, Pride, Print, Klein, Led, Picaro, Zorro, I-Present, EPCBC
FN	DESL/DESXL, TEA/XTEA/XXTEA, Camellia, Simon, SEA, KASUMI, MIBS, LBlock, ITUbee, FeW, GOST, Robin, Fantomas
GFN	CLEFIA, Piccolo, Twis, Twine, HISEC
ARX	Speck, IDEA, HIGHT, BEST-1, LEA
NLFSR	KeeLoq, KATAN/KTANTAN, Halka
Hybrid	Hummingbird, Hummingbird-2, Present-GRP

A. STRUCTURE WISE LWC ALGORITHMS

1) SUBSTITUTION PERMUTATION NETWORK (SPN)

AES [49] is a classic example of SPN based algorithm, standardized by NIST, performs on 128-bit block with 128, 192 and 256-bit key variants [50]. The minimum GE requirement recorded for AES is around 2400 GEs (23% smaller than the usual one) [50], which is still heavy for some small scale real-time applications [35]. It shows the comparatively efficient performance when supplied with additional resources [38].

Another, most hardware and software efficient and ISO/IEC(29192-2P:2012) approved algorithm is **PRESENT**. It is Substitution-Permutation network based, uses 64-bit block on two key variants: 80-bit and 128-bit keys with the GE requirements of 1570 and 1886, respectively [51]. The minimum GE requirement noted for a version of PRESENT is approx. 1000 GE (encryption only) [52], where it takes 2520-3010 GE to provide an adequate level of security [35]. It is a hardware efficient algorithm and uses 4-bit S-boxes (substitution layer - replaces eight S-boxes with single S-box) whereas it takes large cycles in software (permutation layer) which demands an improved version of this [32], [35], [40], [51], [53].

GIFT [54], an improved version of the PRESENT, was presented in CHES-2017. It offers lighter S-Box with smaller physical space. Also, the number of rounds is less and gives high throughput along with the simpler and faster key schedule. There are two versions of GIFT: GIFT-64, 28-round with 64-bit block size and GIFT-128, 40-round with 128-bit block size. Both use a 128-bit key. Also, lighter version, GIFT-64 found more vulnerable than GIFT-128 [55], [56]. Very limited documents have been found with the micro-controller implementation of GIFT [57], [58].

SKINNY [59] has two versions: SKINNY-64 and SKINNY-128. SKINNY-64 uses 64-bit block with 64/128/192-bit key variants to perform 32/36/40 rounds whereas and SKINNY-128 uses 128-bit block with 128/256/384-bit key variants to perform 40/48/56 rounds.

RECTANGLE is an ultra-lightweight block cipher that can be used with various application. With little changes in SPN structure, the rounds are reduced to 25 (compared to 31 rounds in PRESENT) to meet with the competitive environment [53].

TWINE achieves good overall status as PRESENT and also overcomes many of its implementation issues. It operates 64-bit input with two key variants, 80-bit and 128-bit [60]. It requires around 2000 GE and a larger circuit size per throughput compared to AES [12]. In speed comparison, when 1KB or more ROM is available, AES is faster than TWINE, but when only 512bytes of ROM is available, AES can't be implemented and works 250% faster than PRESENT [12].

Midori was designed with a focus on low/tight energy budget, for instance, medical implants. It comes with two different versions, Midori64 and Midori128. Both of these use a 128-bit key on two different block size 64-bit and 128-bit through 16 and 20 iterations, respectively [48], [61].

mCrypton (miniature of Crypton) [62] is a cost and energy-efficient, lightweight edition of Crypton [63], suitable for both hardware and software deployments. It performs 13 iterations on the 64-bit block using a variety of keys (64-bit, 96-bit and 128-bit).

NOEKEON [64] works on the same block and key size, 128 bit, via 16 iterations. The cipher was rejected by the NESSIE project due to its less resistance against the attacks [65].

ICEBERG [66] is optimized for re-configurable hardware deployment with a property of modifying the key at each clock cycle without compromising quality. Here, the round keys are derived on-the-fly. It performs on 64-bit input with 128-bit key via 16 iterations with a demand of 5800 GE at a throughput of 400 Kb/s [67].

PUFFIN-2 [68] is a compact edition of PUFFIN (2303GE) [69]. It uses 80-bit key to perform 34 iterations on 64-bit data using serialized SPN structure. It requires only 1083 GEs for both encryption and decryption.

PRINCE is both hardware and software efficient lightweight algorithm [70] which performs on 64-bit input using a 128-bit key for 12 times [71]. The smallest hardware implementation demands 2953GE at a throughput of 533.3 Kb/s. It shows the low energy consumption of 5.53 μJ/bit [72].

PRIDE [70] exhibits low latency and low energy demand with a 128-bit key to perform 20 iterations on 64-bit input.

PRINT [73] is a domain-specific cipher designed for two applications: PRINT-48 for IC-printing applications which make use of an 80-bit key to perform 48 iterations on 48-bit input (402GE) and PRINT-96 for EPC encryption which uses a 160-bit key to perform 96 iterations on 96-bit input (726GE). It uses 3-bit operations where an odd number of bit operation is not feasible, actual deployment of the algorithm is not ready yet.

Klein [74] works on 64-bit input using 64-bit, 80-bit and 96-bit keys through 12 (1220 GE), 16 (1478 GE), and 20 (1528 GE) iterations, respectively. It was designed with a focus on software implementation, mainly for sensors.

To obtain efficient hardware and software footprints, **LED** [75] borrows features from PRESENT (S-box), Lighter version of AES (row-wise data processing) [50] and PHOTON (mix column approach) [76]. There is an absence of key scheduling in LED which is a unique feature. This approach reduces the chip area but increases the security risk like related key attacks [77]. It processes 64-bit input using various keys such as 64-bit (966 GE), 80-bit (1040 GE), 96-bit (1116 GE) and 128-bit (1265 GE) keys for either 32 or 48 times [75].

PICARO [78] is a novel cipher with a good balance between performance and security (by an adequate choice of S-box). It has 4 different masking levels with faster hardware performance compare to AES. It uses 128-bit key through 12 rounds and shows high resistance to side-channel attacks.

Zorro [79] is based on AES, suitable for embedded systems and more efficient than PICARO. It takes a similar size of block and key (128-bit) through 24 rounds.

EPCBC (Electronic Product Code Block Cipher) [80] is a lightweight cipher, inspired by PRESENT, supports 96-bit key with the input of 48-bit and 96-bit block to perform 32 iterations. The most compact version needs 1008GE. The optimized sub-key generation technique of EPCBC enhances its immunity against related-key differential attacks.

I-PRESENT [81] is an involutive version of PRESENT inspired by PRINCE and NOEKEON. It takes a similar size of the block and key to perform 30 rounds with two additional 4 × 4 S-boxes (16 times). The most compact hardware implementation requires about 2769 GE (encryption and decryption).

2) FEISTEL NETWORK (FN)

The lightweight DES (Data Encryption Standard) is known as **DESL**. It works on a similar size of the block (64-bit), key (56-bit) and a similar number of rounds as DES. The reduced number of S-box (eight to only one [82]) and multiplexer [83] used in DESL distinguishes it from DES. It demands 1850 GE which is 20% compact compare to DES (2310 GE) [83]. DESL also discards the initial and final permutation of DES to make it lighter [84]. **DESXL** is another lighter edition of

DES with a key whitening feature to strengthen the cipher and with 2170 GE demands [83]. It performs the same number of cycles and uses the same block size as DESL but larger key, 184-bit (k = 56, k1 = 64, k2 = 64) [84].

Tiny Encryption Algorithm (TEA) is suitable for very small, computationally weak and low-cost hardware [85]. It operates 128-bit key on 64-bit input to perform 32 rounds [86] with GE requirements of 3872 [87]. Its simple key scheduling is vulnerable to brute force attack [88], [89]. Another limitation of TEA structure is it's three equivalent keys for decryption which makes it vulnerable to the attackers [88]. The improved version of TEA is **(XTEA)** which uses the same size of key and block but with more iterations (64 rounds), demanding 3490 GE [90]. It offers more complex key scheduling with little change in Shift, XOR and addition functions [91]. XTEA was further modified with **XXTEA** [92] to immune against related-key rectangle attack (on 36 rounds) [91].

Camellia [93] is an ISO/IEC, IETF, NESSIE and CRYPTREC recognised cipher. It was designed by Nippon Telegraph and Telephone Corporation and Mitsubishi Electric Corporation. Camellia offers a similar level of security by processing the same size of key and block as AES with two round variants, 18 and 24. It is known for its fast software implementations [94] whereas the hardware implementation requires 6511 GE.

NSA designed **SIMON** [95], which is known for its small footprint in hardware. It offers various keys of size (64-bit, 72-bit, 96-bit, 128-bit, 144-bit, 192-bit, 256-bit) over the block of 32-bit, 48-bit, 64-bit, 96-bit, 128-bit through 32, 36, 42, 44, 52, 54, 68, 69, 72 rounds [95]. The most compact version requires 763GE for execution [95].

SEA [96] is designed for tiny IoT devices, especially for memory-constrained devices [97], with the concept of on-the-fly key generation [96]. It uses 96-bit key on two recommended block size 96-bit and 8-bit with the requirement of 3758GE [97] for the most lightweight hardware version. The optimised software execution demands 426 bytes with encryption cycle of 41604 on 8-bit micro-controllers [98].

KASUMI [99] takes 64-bit input to performs 8 iterations using a 128-bit key. It demands 3437GE for deployment on hardware [100]. It is mainly designed for GSM, UMTS and GPRS systems.

MIBS [101] takes 64-bit input to perform 32 iterations using two variants of keys, 64-bit (1396 GE) and 80-bit (1530 GE). It is Feistel based structure, makes use of S-box from mCrypton [62] and uses PRESENT's keys extraction technique to derive the sub-keys.

LBlock [102] is an ultra-lightweight cipher, performs 32 iterations on 64-bit input along with 80-bit keys. The smallest hardware deployment needs 1320 GE for a throughput of 200 Kb/s whereas 3955 clock cycles are taken by most efficient software implementation to encrypt a single block (on the 8-bit microcontroller).

The designed and developed by the government of the Soviet Union (1989), the lightweight version of **GOST** executes on 64-bit input with a 256-bit key for 32 times. The S-Box in this version is adopted from PRESENT [103] with the demands of 651 GE.

ITUbee [104] is a software efficient cipher with a code size of 586 bytes and 2937 cycles (the most compact version of encryption). It takes the same size of key and block (80-bit). Here, key scheduling is replaced by round-dependent constants to reduce software overload.

FeW [105] processes 64-bit input with two varieties of the key, 80-bit and 128-bit for 32 times. It makes use of S-box of Humminbird-2 and follows the key expansion process from the PRESENT. There no cryptanalytic attack found on FeW [105].

3) GENERALISED FEISTEL NETWORK (GFN)

Introduced by SONY corporation and approved by NIST, **CLEFIA** offers 128-bit block with choice of 128, 192, 256 bit key through 18, 22, 26 round, respectively [106], [107]. It shows high performance and strong immunity against various attacks [40], [106] [108], [109] with comparative high cost as the most compact version requires 2488 GE (encryption only) for 128-bit key [107]. The strong immunity of CLEFIA against security attacks is grateful to its dual confusion and diffusion properties. In contrary, this demands higher memory and limits its use in ultra-small applications [35].

Piccolo [110] is another ultra-lightweight cryptography algorithm suitable for extremely restricted environmental devices (RFID, sensors, etc.). It processes 64-bit input to perform two iterations, 25 and 31, using two key sets, 80-bit and 128-bit, respectively. The smallest hardware deployment (80-bit key) requires 432 GE and an additional 60 GE to perform decryption.

TWIS [111], derived from CLEFIA, takes equal size block and key (128-bit) to perform 10 iterations. It is a victim of differential distinguisher with probability one [112].

TWINE [60], derived from LBlock, performs 36 iterations on 64-bit state along with two key options, 80-bit and 128-bit. The most compact hardware implementation requires 1866 GE. TWINE uses nibble permutation instead of bit permutation (for sub-key generation) of LBlock. Also, it uses a single S-box instead of ten S-Boxes of LBlock.

HISEC [113] performs 15 iterations on 64-bit input along with an 80-bit key, demanding 1695 GE. It shows good resistance against different attacks, and the characteristics are more like to PRESENT except bit-permutation.

4) ADD-ROTATE-XOR (ARX)

SPECK [95], sibling of SIMON and designed by NSA, is a software-oriented cipher. It supports the similar size of blocks and keys as SIMON to perform 22, 23, 26, 27, 28, 29, 32, 33 and 34 iterations. The most compact hardware implementation recorded uses 48-bit block with 96-bit key with requirements of 884 GE whereas the most efficient software implementation requires 599 cycles with 186-byte of ROM for 64-bit block with 128-bit key [95].

IDEA [114], designed by Lai and Massey, makes use of a 128-bit key on 64-bit input to perform 8.5 iterations, mainly used for high-speed networks [115]. It uses 16-bit unsigned integer and performs data operations such as XOR, addition and modular multiplication without using S-box or P-box. It is known for its best performance on embedded systems (such as PGP v2.0.) with memory needs of 596 bytes at a throughput of 94.8 Kb/s (the smallest software version) [116].

HIGHT [117], an ultra-lightweight algorithm, processes 64-bit data using a 128-bit key for 32 times. It performs compact round function (no S-boxes) using simple computational operations. The most compact version acquires 2608 GE for 188 Kbps throughput [118].

BEST-1 [119], an ultra-lightweight cipher, targets Wireless Sensor Networks and RFID tags. It takes 64-bit input with a 128-bit key through 12 rounds on 8-bit processors, demanding 2200 GE. The core functions of BEST-1 are mod 2^8 addition and subtraction, bitwise shift and XOR.

LEA [120] is a software-oriented cipher and was introduced by the ETRIK for 32-bit common processor. It processes 128-bit input to perform 24, 28, and 32 iterations using 128-bit, 192-bit and 256-bit keys, respectively. On the ARM platform, LEA performs 326.94 cycles/byte with a storage demand of 590 bytes (code) and 32 bytes for execution. The most compact version requires 3826 GE for 76.19 Mbps throughput [121].

5) NONLINEAR-FEEDBACK SHIFT REGISTER (NLFSR)

With focus on automobile industry, **KeeLoq** [22] is designed with an aim to keyless authentication (remote access) in cars [122] by Gideon Kuhn. It takes 32-bit input with a 64-bit key to perform 528 rounds. Even though KeeLoq was developed in the '80s, the cryptanalysis report was issued in February 2007 for the first time by Bogdanov [123].

KATAN/KTANTAN [124], inspired by KeeLoq, cipher family applies 80-bit key on various block size (32-bit, 48-bit and 64-bit) through 254 iterations. They could be executed on small-scale hardware (KATAN 802 GE and KTANTAN 462 GE), as mainly designed for RFID tags and sensor networks. They follow a linear structure (LFSR) instead of NLFSR of KeeLoq. KATAN has a very simple key scheduling compare to KeeLoq, whereas KTANTAN exhibits no key generation operations (reduce GE requirement). As the key remains unchanged once initialized, the applications of KTANTAN is limited. KTANTAN-48 (588 GE) is more appropriate for RFID tags. In software, both shows poor performance (low throughput and high energy consumption) due to overuse of bit manipulation [98].

Halka [125] performs well on both hardware and software. It takes 64-bit input with an 80-bit key to perform 24 iterations. The multiplicative inverse based S-boxes (8-bit) with LFSR makes Halka more secure than PRESENT. It demands 138 GE (7% less GE than PRESENT) [125]. Also, the software performance is 3 times more efficient than PRESENT [125].

6) HYBRID

Hummingbird [126] is an ultra-lightweight algorithm, introduces a hybrid structure (block and stream). It takes 16-bit input with a 256-bit key to perform 20 iterations. It was vulnerable to several attacks [127].

Hummingbird-2 [128], designed for low-end microcontrollers, takes 64-bit input (initial vector) with a 128-bit key. It performs well on both the platforms (hardware/software). It also satisfies the ISO 18000-6C protocol. It gives better performance compare to PRESENT (on 4-bit microcontrollers) but have few drawbacks: 1) Initialization is necessary before encryption (or decryption) due to its stream property 2) Different encryption and decryption functions and due to that full version is 70% heavier than only encryption. Moreover, its performance degrades while processing small messages.

PRESENT-GRP [35] works on 64-bit input with a 128-bit key to perform 31 iterations. It makes use of the substitution-permutation technique from PRESENT along with a group(GRP) operation for additional confusion properties (in replacement of permutation table). The hardware implementation of PRESENT (1884 GE) is slightly better than PRESENT-GRP (2125 GE). Similarly, PRESENT is more efficient than PRESENT-GRP in software implementation too.

B. HARDWARE AND SOFTWARE PERFORMANCE COMPARISON

Various experiments have been carried out by many researchers using different platforms such as NXP [35], AVR [129], ARM [35] micro-controllers to evaluate the performance of the popular lightweight cryptography algorithms [35], [38] [13], [40] [50], [83] [129], [130]. During these experiments, various characteristics such as area (GE), logic process (μm), power consumption (μW), throughput, RAM/ROM requirements (*bytes*), latency (*cycle/block*), etc. have been compared for different lightweight cryptography algorithms in different circumstances (file types (C/C++, Java, Python), message size, etc.). Table 4 summarizes the hardware and software performance of the listed LWC algorithms evaluated on 0.09/0.13/0.18/0.35 μm technologies (hardware implementation) and on 8/16/32 bit micro-controllers (software implementation) platforms.

According to the graph (Figure 5), software efficiency competition is won by SPECK, followed by SIMON and then PRIDE. Also, ITUbee, LEA, IDEA and AES show better software efficiency compare to the other LWC algorithms.

Memory (RAM and ROM) requirements by various LWC algorithms can be studied from the above graph (Figure 6) which reveals the first ten, most memory-efficient LWC algorithms. The competition is again won by SPECK and SIMON with less than 200 bytes of ROM and zero bytes of RAM requirement, closely followed by PRIDE.

Another important software metrics, latency and throughput, lead by again SPECK and SIMON with lowest latency rate (408 and 594 cycles/block) and highest throughput

TABLE 4. Hardware and Software performances of LWC algorithms.

LWC Algorithm	Hardware Implementation								Software Implementation							
	Key Size	Block Size	Tech (μm)	Area (GEs)	Power (μW)	Energy (μJ/bit)	Throughput @100KHz (Kbps)	Hardware Efficiency (Kbps/KGE)	Key Size	Block Size	ROM (byte)	RAM (byte)	Latency (Cycles/ block)	Energy (μJ/bit)	Throughput @4MHz (Kbps)	Software Efficiency (Kbps/KB)
AES*	128	128	0.13	2400	2.4	42.38	56.64	23.6	128	128	918	0	4192	16.7	122	132.9
PRESENT*	80	64	0.18	1570	2.35	11.77	200	127.38	128	64	660	0	10792	43.1	23.7	35.91
RECTANGLE	80	64	0.13	1467	1.46	5.96	246	167.68	-	-	-	-	-	-	-	-
MIDORI	128	64	0.09	1542	60.6	1.61	400	259.4	-	-	-	-	-	-	-	-
mCrypton*	128	64	0.35	2594	4.66	138.61	33.51	12.91	96	64	1076	28	16457	68	15.5	14.41
NOEKEON*	128	128	0.35	2604	4.68	1362.21	3.44	1.32	128	128	364	32	23517	95.9	21.7	59.62
ICEBERG	128	64	0.18	5817	8.72	21.81	400	68.76	-	-	-	-	-	-	-	-
PUFFIN-2	80	64	0.18	1083	1.62	314.74	5.2	4.8	-	-	-	-	-	-	-	-
PRINCE*	128	64	0.13	2953	2.95	5.53	533.3	180.59	128	64	1108	0	3614	14.4	70.8	63.9
PRIDE*	-	-	-	-	-	-	-	-	128	64	266	0	1514	6	169	635.34
PRINT^	80	48	0.18	503	0.75	7.54	100	198.8	80	48	6210	48	87272	117.8	2.2	0.35
Klein*	64	64	0.18	1220	1.83	59.18	30.9	25.32	64	64	2980	50	7901	10.6	32.4	10.87
LED#	64	64	0.18	966	1.45	282.55	5.1	5.27	80	64	2164	368	35161	-	7.28	3.36
I-PRESENT	80	64	0.18	2467	370	-	-	-	-	-	-	-	-	-	-	-
EPCBC	96	48	0.18	1008	1.51	124.74	12.12	12.02	-	-	-	-	-	-	-	-
DESL*	56	64	0.18	1848	2.77	62.37	44.4	24.02	56	64	3098	0	8365	33.4	30.6	9.88
TEA*	128	64	0.18	2355	3.53	35.32	100	42.46	128	64	648	24	7408	30.3	34.5	53.24
XTEA*	-	-	-	-	-	-	-	-	128	64	504	0	17514	70	14.6	28.97
Camellia*	128	128	0.18	6511	9.76	33.57	290.1	44.55	128	128	1262	12	64000	256	8	6.34
SIMON*	96	48	0.13	763	0.76	48.32	15.8	20.7	96	48	170	0	594	2.3	323	1900
SEA*	96	8	0.13	2562	2.56	1117.67	2.29	0.89	96	96	426	24	41604	173.7	9.2	21.6
KASUMI*	128	64	0.13	3437	3.44	29.9	115.14	33.5	128	64	1264	24	11939	47.6	21.4	16.93
MIBS^	64	64	0.18	1396	2.09	10.47	200	143.26	64	64	3184	29	49056	66.2	5.2	1.63
LBlock^	80	64	0.18	1320	2	9.9	200	151.51	80	64	976	58	18988	25.6	13.48	13.81
ITUbee*	-	-	-	-	-	-	-	-	80	80	716	0	2607	10.4	122.7	171.37
GOST^	256	64	0.18	1000	1.5	7.5	200	200	256	64	4748	190	10240	13.8	25	5.27
Robin^	-	-	-	-	-	-	-	-	128	128	1942	80	4935	6.6	103.74	53.42
Fantomas^	-	-	-	-	-	-	-	-	128	128	1920	78	3646	4.9	140.42	73.14
CLEFIA*	128	128	0.13	2678	2.67	36.82	76	28.37	128	128	3046	0	28648	114.5	17.8	5.84
PICCOLO^	80	64	0.13	1136	1.13	4.8	237.04	208.66	80	64	966	70	21448	28.9	11.93	12.35
TWINE#	80	64	0.09	1503	1.05	5.91	178	118.42	80	64	1180	140	20505	-	12.48	10.58
SPECK*	96	48	0.13	884	0.88	73.67	12	13.57	96	48	134	0	408	1.6	470.5	3511.19
IDEA*	-	-	-	-	-	-	-	-	128	64	596	0	2700	10.8	94.8	159.06
HIGHT*	128	64	0.35	2608	4.7	24.93	188	72.08	128	64	5718	47	6377	25.5	40.14	7.02
LEA#	128	128	0.13	3826	3.82	50.22	76.19	19.91	128	128	590	32	5231	-	97.8	165.76
KATAN*	80	32	0.13	802	0.8	64.16	12.5	15.58	80	64	338	18	72063	289.2	3.5	10.35
KTANTAN^	80	32	0.13	462	0.46	36.96	12.5	27.05	80	32	10516	614	10233211	13814.8	0.012	0
Hummingbird^	-	-	-	-	-	-	-	-	128	16	1822	82	4637	6.2	13.8	7.57
Hummingbird-2^	128	16	0.18	2159	3.23	40.48	80	37.05	128	16	770	50	1520	2	42.1	54.68

* 8-bit microcontroller, ^16-bit microcontroller, # 32-bit microcontroller

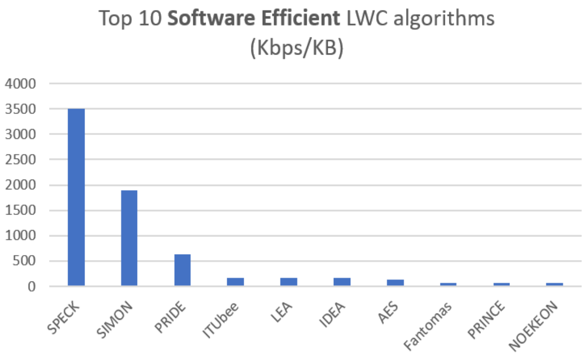


FIGURE 5. Software Efficient LWC algorithms (Top 10).

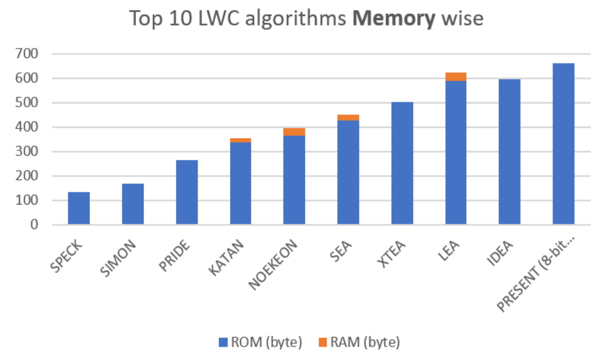


FIGURE 6. Memory Efficient LWC algorithms (Top 10).

(470.5 and 323 Kb/s) unceasingly followed by PRIDE. ITUbee and IDEA also secure their places in the list of first ten performers (Figure 7).

In terms of hardware efficiency, Midori is on the top of the list, by PICCOLO as runners-up with a minor difference with GOST. Figure 8 visualizes the first ten hardware efficient LWC algorithms.

SEA leads the key and block wise hardware efficiency competition with very little block size (only 8-bit), followed by Hummingbird-2 with a double-size block (and the largest key in this top-10 list) and further by KATAN/KTANTAN with 4 times bigger block compared to the leader (Figure 9). The list accommodates PRINT, EPCBC, SIMON/SPECK, PRESENT and RECTANGLE with either 48-bit or 64-bit block along with 80-bit or 96-bit key.

From the graph (Figure 10), we can say that KTANTAN demands the smallest area (462 GE) to implement, with a minor difference from PRINT (41 GE more). SPECK/SIMON shows their presence in top 5 lists with less than 900 GE needs. All of these performances are noticed either on 0.13 μm or 0.18 μm technologies.

In terms of energy consumption, Midori shows the lowest energy requirement (1.61μJ/bit), followed by Piccolo,

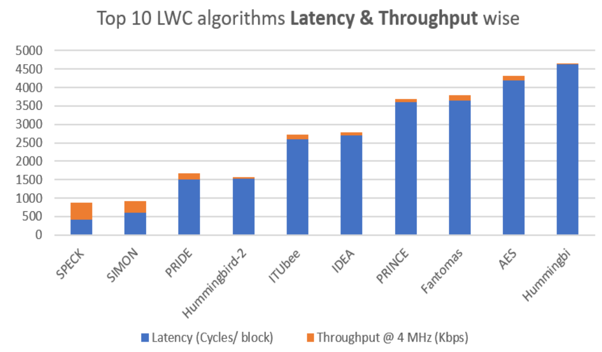


FIGURE 7. Latency Efficient LWC algorithms (Top 10).

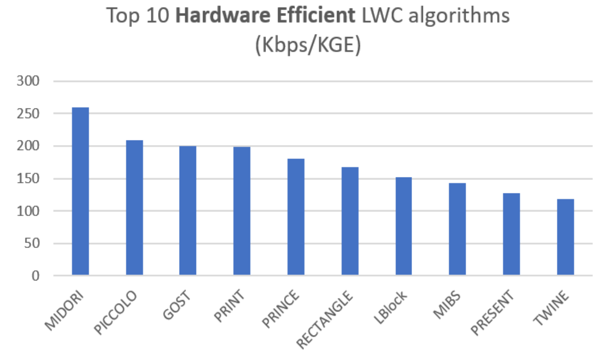


FIGURE 8. Hardware Efficient LWC algorithms (Top 10).

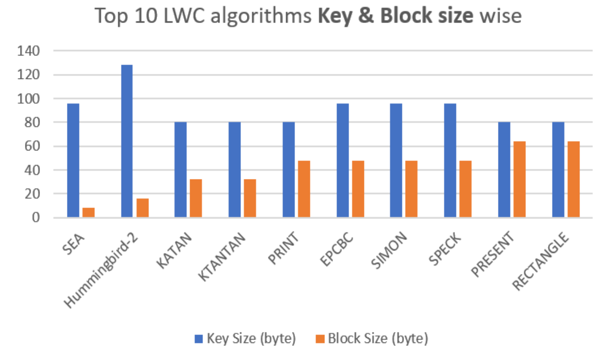


FIGURE 9. Key & Block size wise Hardware Efficient LWC algorithms (Top 10).

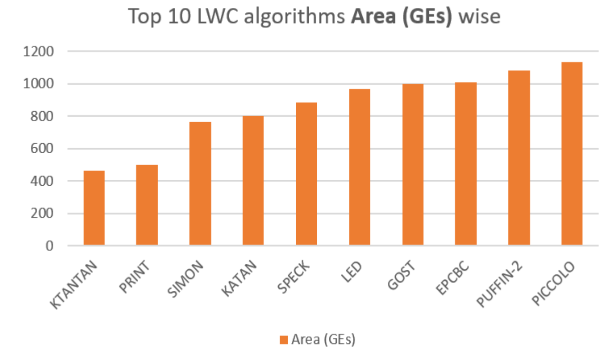


FIGURE 10. Physical Area wise Hardware Efficient LWC algorithms (Top 10).

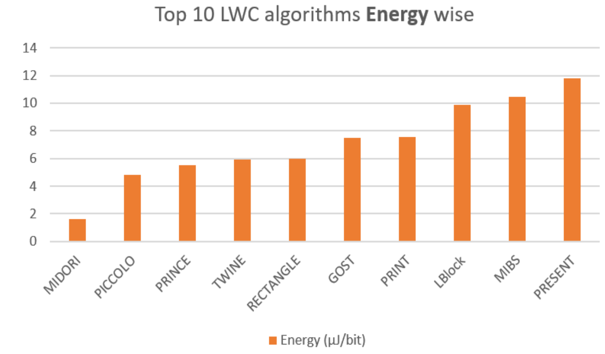


FIGURE 11. Energy Efficient Hardware Efficient LWC algorithms (Top 10).

PRINCE, TWINE and RECTANGLE with small differences amongst (Figure 11).

In summary, SIMON and SPECK shine by their most efficient software implementation but disappears from the top-10 list of hardware efficient LWC algorithms. Also, derived version of AES such as PRESENT and derived lighter versions of DES such as DESL/DESLX, CLEFIA are widely recognised algorithms (by the standardising bodies) due to high-security reasons. Overall, none of the LWC algorithms meets all the efficiency metrics of the hardware and software requirements and shows distinct performances in different circumstances.

C. CRYPTANALYSIS OF LWC ALGORITHMS

Along with performance and cost, security is an important and essential measure for any lightweight cryptography algorithm. Attack resistance property of any lightweight cryptography algorithm can be measured through cryptanalysis. Cryptanalysis aims at detecting algorithm vulnerabilities by attempting various attacks and decryption techniques [38]. The main 4 types of cryptanalysis on block cipher are [38], [51], [53], [131]: Differential cryptanalysis, Linear cryptanalysis, Integral cryptanalysis and Algebraic cryptanalysis. **Differential cryptanalysis** is an analysis of outputs against various inputs. The special types are higher-order, truncated, impossible and boomerang. **Linear cryptanalysis** postulates a linear approximation based on the piling-up lemma principle (introduced by Mitsuru Matsui) between plaintext, ciphertext and key by characters or individual bits. **Integral cryptanalysis** is especially pertinent to block ciphers with substitution-permutation networks. It is documented with two other names such as Square attack and saturation attack too. **Algebraic cryptanalysis** is based on equation-solving algorithms and has been proven effective on lightweight versions due to its simple structure (less number of rounds with less algebraic complexity).

These cryptanalyses are based on Ciphertext only, Known plaintext, Chosen plaintext and Chosen ciphertext along with MITM, Brute force and side channel. Differential Fault Attacks, a type of side-channel attack, analyzes the internal structure and finds an exploitable place to attack the algorithm [132], [133]. **Table 5** demonstrates the security analysis of various LWC algorithms in a grid form. The study shows that almost all existing lightweight block cipher solutions suffer from various attacks, especially, related-key attack, followed by various differential and MITM attacks. Moreover, the lighter versions (with reduced rounds) are more vulnerable to various attacks compared to their standard one.

D. STANDARDIZATION OF LWC ALGORITHMS

The organizations/research groups, who are actively contributing in the field of cryptography to improve the lightweight standards for resource-constrained devices are s follows:

- National Institute of Standards and Technology, USA (NIST)
- International Organization of Standardization and the International Electrotechnical Commission (ISO/IEC)
- Cryptography Research and Evaluation Committees, Japan (Cryptrec)
- European Network of Excellence in Cryptology (Ecrypt)
- National Security Agency of USA (NSA)
- CryptoLUX (University of Luxembourg)

PRESENT [51] and CLEFIA [106] are the only two algorithms approved by the ISO/IEC 29192 standard whereas AES, CLEFIA, TDES, Camellia, PRESENT, PRINCE, Piccolo, LED, TWINE, SIMON & SPECK, Midori are targeted by Cryptrec.

E. REAL-TIME USE CASES: APPLICATIONS & THEIR LIGHTWEIGHT DEMANDS

The wide range of IoT applications in various fields creates the demand for lightweight cryptography algorithms with different requirements [174]. **Smart home appliances** such as smart TV, smart fridge, smart kettle, smart bulbs, etc., demands for small memory and small processing. The best suit algorithms in this scenario are SIMON, SPECK, PICCOLO and TWINE. Due to tiny physical space and a little or no power backup in RFID tags, SIMON, SPECK, Piccolo and PRINCE are the best options for **logistics applications**. Nowadays, **smart agriculture** is an emerging field that demands compact implementation, less processing cycles, little power consumption with plenty of sensors in a remote location. SIMON, SPECK, PRESENT and TWINE fulfil the requirements of smart agriculture. A person under medical treatment in a hospital or at a residence could be monitored for pulse count, level of pressure, sugar and oxygen in the blood, using IoT sensors where security and privacy of the transmitting data are crucial along with tiny circuitry, little processing power and limited batteries (in case of an implanted device) and quick response time. In this constrained environment, SIMON, SPECK, PICCOLO, PRESENT and Midori are the best suit solutions to secure the communication in **health care applications** due to their overall compact hardware and software implementation to match with a real-time response while in-body and/or out-body (wearable) implantation. For **industrial systems (Industry 4.0)** where sensors could be attached to equipment at various places (not easily accessed by the operators), to transmit the data wirelessly for specific distances. In this state, real-time processing is the key element with adequate security (without bothering about energy consumption). Midori and PRINCE show the best performance in a demanding scenario. In an era of 5G technology, **automobile industry** demands not only in-vehicle communication but also with infrastructures such as traffic signals and road signs (V2X). This communication demands a prompt response (low latency) on a tiny circuitry with high security. Midori, PRINCE, PRESENT, and SIMON are the right choices for auto industries. Keeloq is another powerful LWC algorithm for secure remote keyless entry in cars and buildings [171].

£
Cost
Performance
Security

FIGURE 12. **Cost, Performance and Security.**

IV. OPEN RESEARCH CHALLENGES AND RESEARCH DIRECTIONS

The ideal algorithm should maintain a proper balance among cost, performance and security (Figure 12). Any two of these three can be easily optimized, whereas achieving all of these together is challenging [38]. For example, an increasing number of rounds [131] or key size results in degradation of algorithm performance. These could be achieved by design focus on less memory and less computing power requirement, leading to less Gate Equivalent (physical area) requirements along with low power (energy) consumption without compromising strong security [35]. Based on the above study, we have identified the following research issues, which require further attention to make the LWCs algorithms effective in IoT security:

1) One of the two fundamental properties of cryptography, confusion, could be achieved by choosing an efficient and adequate number of S-boxes to demonstrate a proper balance between performance and security [78]. So designing simple and fast but strong confusion (Substitution, S-box) and diffusion (Bit Permutation) properties with right balance amongst cost, performance and security is of practical interest, e.g., How to reduce

TABLE 5. Security Analysis of LWC Algorithms.

LWC Algorithm	Differential Cryptanalysis*	Linear Crypt-analysis	Integral/Square/ Saturation Cryptanalysis	Algebric/Cube Cryptanalysis	MITM/ Biclique	Related Key attack	Side-Channel/ Differential fault attacks
AES	✓ [134]	-	-	-	✓ [50]	✓ [50]	✓ [135] [136] [137]
PRESENT	✓ [138] [139]	-	-	-	✓ [28]	✓ [140]	✓ [135] [137] [141] [142]
GIFT	✓ [143] [56]	-	✓ [144]	-	✓ [54] [144] [57]	✓ [145] [56] [146]	✓ [147]
SKINNY	✓ [148]	-	-	-	-	✓ [146]	✓ [137] [149] [150]
RECTANGLE	-	-	✓ [53]	-	-	✓ [53]	✓ [53]
MIDORI	✓ [48]	✓ [48]	-	-	✓ [48]	-	-
mCrypton	-	-	-	-	-	✓ [151]	-
NOEKEON	-	-	-	-	-	✓ [65]	-
ICEBERG	✓ [152]	-	-	-	-	-	-
PUFFIN-2	✓ [153]	-	-	-	-	-	-
PRINCE	✓ [154]	-	-	-	-	-	-
PRINT	-	-	-	-	-	✓ [155]	-
Klein	-	-	-	-	✓ [156]	✓ [157]	✓ [158]
LED	✓ [159]	-	-	-	✓ [28]	✓ [77]	-
EPCBC	-	-	-	✓ [23]	-	✓ [23]	-
TEA	-	-	-	-	-	✓ [88] [89]	-
XTEA	-	-	-	-	-	✓ [91]	-
XXTEA	✓ [160]	-	-	-	-	-	-
Camellia	✓ [94]	-	-	-	-	-	✓ [24] [136]
SIMON	✓ [29] [161]	-	-	✓ [30]	-	✓ [30]	-
KASUMI	✓ [17]	-	-	-	-	✓ [18]	-
MIBS	✓ [162]	✓ [162]	-	-	-	-	-
LBlock	✓ [25] [26]	-	✓ [60]	-	✓ [163]	-	-
ITUbee	-	-	-	-	-	✓ [164]	-
GOST	-	-	-	-	✓ [165]	✓ [166]	-
CLEFIA	-	-	✓ [107]	-	-	✓ [107]	✓ [136]
PICCOLO	✓ [167]	-	-	-	✓ [28] [168]	-	-
TWIS	✓ [112]	-	-	-	-	-	-
TWINE	-	-	✓ [84]	-	✓ [169]	-	-
SPECK	✓ [161] [131]	-	-	-	-	✓ [131]	-
IDEA	-	-	-	-	✓ [16]	-	-
HIGHT	✓ [140]	✓ [15]	-	-	✓ [168]	✓ [170]	-
LEA	-	-	-	-	-	-	✓ [27]
KeeLoq	-	✓ [123]	-	✓ [122]	✓ [171]	-	✓ [123]
KATAN	-	-	-	-	✓ [172]	-	-
KTANTAN	-	-	-	-	-	✓ [19]	-
Hummingbird-2	-	-	-	-	-	✓ [173]	-
Hummingbird	Vulnerable to several attacks [127]						

*It includes Truncated/Higher-order/Impossible/Boomerang Differencial Cryptanalysis

the number of S-boxes as they increase the demands for memory (to store) and computing power (to produce) while maintaining the same security level? (motivation: PRESENT is designed from AES and replaces eight S-boxes with just one. Similarly, many researchers have derived the lighter versions from the standard cryptography algorithms with a few modifications by reducing substitution-permutation (counter-effect on security level)). But how to replace S-boxes with some other confusion techniques with the same level of security and less overhead of memory and processing cost is still an open problem.

2) Making key scheduling lighter with smaller key size and adequate strength, i.e., How to generate random sub-keys from the provided initial key for all n rounds?
3) Increase in the number of rounds adversely affects the performance and cost, i.e., How to decrease (or increase) number of rounds without compromising performance as well as security level?

We are currently working on substitution-permutation methods with main focus on S-Box to design a generic lightweight cryptography algorithm, with the right blend of three main characteristics namely, cost, performance and security.

V. CONCLUSION

Due to the exponential growth in the number of IoT devices in various domains, IoT security is one of the main concerns. As a consequence, there is a need for a lightweight algorithm(s) with trade-offs amongst cost and performance and security. For resource-constrained IoT devices, lightweight cryptography is an effective way to secure communication by transforming the data. The well-defined LWC characteristics (cost, performance and security) by NIST are compared, and further research gaps and open research challenges are highlighted in this paper. From the literature review, PRESENT and CLEFIA are the approved block ciphers by NIST due to security reasons along with accepted performance and cost.

On the other side, SIMON and SPECK impress by their most compact implementations. In general, none of the LWC algorithms fulfils all the criteria of hardware and software performance metrics but performs at their best in the specified environment. However, new attacks are reported with the growth of new LWC algorithms which is an inevitable and never-ending process. The war between cybersecurity experts and attackers always opens a door of opportunities for new research in the field of cybersecurity, especially lightweight cryptography.

REFERENCES

[1] G. Kortuem, F. Kawsar, V. Sundramoorthy, and D. Fitton, "Smart objects as building blocks for the Internet of Things," *IEEE Internet Comput.*, vol. 14, no. 1, pp. 44–51, Jan. 2010.

[2] N. P. Moldón, "Security in IoT ecosystems," Univ. Oberta de Catalunya (UOC), Barcelona, Spain, Tech. Rep. 10609/97707, 2016. [Online]. Available: http://hdl.handle.net/10609/97707

[3] E. Brown, *21 Open Source Projects For IoT*, vol. 23. Linux.com, 2016. [Online]. Available: https://www.linux.com/news/21-open-source-projects-iot/

[4] S. Charmonman and P. Mongkhonvanit, "Internet of Things in E-business," in *Proc. 10th Int. Conf. E-Bus. King Mongkut's Univ. Technol. Thonburi*, 2015, pp. 1–9.

[5] (Aug. 2015). *The Trouble With the Internet of Things*. [Online]. Available: https://data.london.gov.uk/blog/the-trouble-with-the-internet-of-things

[6] K. McKay, L. Bassham, M. S. Turan, and N. Mouha, *Report on Lightweight Cryptography (Nistir8114)*. Gaithersburg, MD, USA: NIST, 2017.

[7] B. J. Mohd and T. Hayajneh, "Lightweight block ciphers for IoT: Energy optimization and survivability techniques," *IEEE Access*, vol. 6, pp. 35966–35978, 2018.

[8] A. Banafa, "Three major challenges facing IoT," *IEEE IoT Newslett.*, Mar. 2017. [Online]. Available: https://iot.ieee.org/newsletter/march-2017/three-major-challenges-facing-iot.html

[9] S. Singh, P. K. Sharma, S. Y. Moon, and J. H. Park, "Advanced lightweight encryption algorithms for IoT devices: Survey, challenges and solutions," *J. Ambient Intell. Hum. Comput.*, vol. 4, pp. 1–18, May 2017.

[10] W. Feng, Y. Qin, S. Zhao, and D. Feng, "AAoT: Lightweight attestation and authentication of low-resource things in IoT and CPS," *Comput. Netw.*, vol. 134, pp. 167–182, Apr. 2018.

[11] B. J. Mohd, T. Hayajneh, and A. V. Vasilakos, "A survey on lightweight block ciphers for low-resource devices: Comparative study and open issues," *J. Netw. Comput. Appl.*, vol. 58, pp. 73–93, Dec. 2015.

[12] O. Toshihiko, "Lightweight cryptography applicable to various IoT devices," *NEC Tech. J.*, vol. 12, no. 1, pp. 67–71, 2017.

[13] A. Biryukov and L. P. Perrin, "State of the art in lightweight symmetric cryptography," Univ. Luxembourg Library, Esch-sur-Alzette, Luxembourg, Tech. Rep. 10993/31319, 2017. [Online]. Available: https://orbilu.uni.lu/handle/10993/31319

[14] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. Mccann, and K. Leung, "A survey on the ietf protocol suite for the Internet of Things: Standards, challenges, and opportunities," *IEEE Wireless Commun.*, vol. 20, no. 6, pp. 91–98, Dec. 2013.

[15] L. Wen, M. Wang, A. Bogdanov, and H. Chen, "Multidimensional zero-correlation attacks on lightweight block cipher HIGHT: Improved cryptanalysis of an ISO standard," *Inf. Process. Lett.*, vol. 114, no. 6, pp. 322–330, Jun. 2014.

[16] D. Khovratovich, G. Leurent, and C. Rechberger, "Narrow-bicliques: Cryptanalysis of full idea," in *Proc. 31st Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, Apr. 2012, pp. 392–410. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-29011-4_24

[17] E. Biham, O. Dunkelman, and N. Keller, "A related-key rectangle attack on the full KASUMI," in *Proc. 11th Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, Dec. 2005, pp. 443–461. [Online]. Available: https://link.springer.com/chapter/10.1007/11593447_24

[18] T. Saito, "A single-key attack on 6-round KASUMI," in *Proc. IACR*, Dec. 2011, p. 584.

[19] M. Ågren, "Some instant-and practical-time related-key attacks on ktantan32/48/64," in *Proc. 18th Int. Workshop Sel. Areas Cryptogr. (SAC)*. Berlin, Germany: Springer-Verlag, Aug. 2011, pp. 213–229. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-28496-0_13

[20] A. Bogdanov, "Cryptanalysis of the KeeLoq block cipher," in *Proc. IACR*, 2007, p. 55.

[21] N. T. Courtois, G. V. Bard, and D. Wagner, "Algebraic and slide attacks on Keeloq," in *Proc. 15th Int. Workshop Fast Softw. Encryption (FSE)*. Berlin, Germany: Springer, Feb. 2008, pp. 97–115.

[22] S. Indesteege, N. Keller, O. Dunkelman, E. Biham, and B. Preneel, "A practical attack on Keeloq," in *Proc. 27th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, Apr. 2008, pp. 1–18. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-78967-3_1

[23] M. Walter, S. Bulygin, and J. Buchmann, "Optimizing guessing strategies for algebraic cryptanalysis with applications to EPCBC," in *Proc. 8th Int. Conf. Inf. Secur. Cryptol.* Berlin, Germany: Springer, Nov. 2012, pp. 175–197. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-38519-3_12

[24] X.-J. Zhao, T. Wang, and Y. Zheng, "Cache timing attacks on camellia block cipher," in *Proc. IACR*, 2009, p. 354.

[25] K. Jeong, C. Lee, and J. I. Lim, "Improved differential fault analysis on lightweight block cipher LBlock for wireless sensor networks," *EURASIP J. Wireless Commun. Netw.*, vol. 2013, no. 1, p. 151, Dec. 2013.

[26] H. Yoshikawa, M. Kaminaga, A. Shikoda, and T. Suzuki, "Secret key reconstruction method using round addition dfa on lightweight block cipher lblock," in *Proc. Int. Symp. Inf. Theory Appl.*, 2014, pp. 493–496.

[27] Y. Kim and H. Yoon, "First experimental result of power analysis attacks on a FPGA implementation of LEA," in *Proc. IACR*, 2014, p. 999.

[28] K. Jeong, H. Kang, C. Lee, J. Sung, and S. Hong, "First experimental result of power analysis attacks on a FPGA implementation of LEA," in *Proc. IACR*, 2012, p. 621.

[29] H. AlKhzaimi and M. M. Lauridsen, "Cryptanalysis of the Simon family of block ciphers," in *Proc. IACR*, 2013, p. 543.

[30] R. Rabbaninejad, Z. Ahmadian, M. Salmasizadeh, and M. R. Aref, "Cube and dynamic cube attacks on SIMON32/64," in *Proc. 11th Int. ISC Conf. Inf. Secur. Cryptol.*, Sep. 2014, pp. 98–103.

[31] F. Abed, E. List, S. Lucks, and J. Wenzel, "Differential and linear cryptanalysis of reduced-round Simon," Citeseer, Cryptol. ePrint Arch., Tech. Rep. 2013/526, 2013. [Online]. Available: https://eprint.iacr.org/2013/526.pdf

[32] I. Bhardwaj, A. Kumar, and M. Bansal, "A review on lightweight cryptography algorithms for data security and authentication in IoTs," in *Proc. 4th Int. Conf. Signal Process., Comput. Control (ISPCC)*, Sep. 2017, pp. 504–509.

[33] W. Diehl, F. Farahmand, P. Yalla, J.-P. Kaps, and K. Gaj, "Comparison of hardware and software implementations of selected lightweight block ciphers," in *Proc. 27th Int. Conf. Field Program. Log. Appl. (FPL)*, Sep. 2017, pp. 1–4.

[34] N. Hanley and M. ONeill, "Hardware comparison of the ISO/IEC 29192-2 block ciphers," in *Proc. IEEE Comput. Soc. Annu. Symp.*, Aug. 2012, pp. 57–62.

[35] G. Bansod, N. Raval, and N. Pisharoty, "Implementation of a new lightweight encryption design for embedded security," *IEEE Trans. Inf. Forensics Security*, vol. 10, no. 1, pp. 142–151, Jan. 2015.

[36] S. Kerckhof, F. Durvaux, C. Hocquet, D. Bol, and F.-X. Standaert, "Towards green cryptography: A comparison of lightweight ciphers from the energy viewpoint," in *Proc. 14th Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, Sep. 2012, pp. 390–407. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-33027-8_23

[37] A. Shah and M. Engineer, "A survey of lightweight cryptographic algorithms for iot-based applications," in *Proc. Smart Innov. Commun. Comput. Sci.* Singapore: Springer, 2019, pp. 283–293, doi: 10.1007/978-981-13-2414-7.

[38] S. Sallam and B. D. Beheshti, "A survey on lightweight cryptographic algorithms," in *Proc. IEEE Region Conf.*, Oct. 2018, pp. 1784–1789.

[39] C. G. Thorat and V. S. Inamdar, "Implementation of new hybrid lightweight cryptosystem," *Appl. Comput. Informat.*, vol. 16, nos. 1–2, pp. 195–206, May 2018.

[40] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, and C. Manifavas, "A review of lightweight block ciphers," *J. Cryptograph. Eng.*, vol. 8, no. 2, pp. 141–184, 2018.

[41] K. Mohajerani, R. Haeussler, R. Nagpal, F. Farahmand, A. Abdulgadir, J.-P. Kaps, and K. Gaj. (2020). *FPGA Benchmarking of Round 2 Candidates in the Nist Lightweight Cryptography Standardization Process: Methodology, Metrics, Tools, and Results*. [Online]. Available: https://eprint.iacr.org/2020/1207

[42] C. L. C. W. Group *et al.*, "Creptrec cryptographic technology guideline (lightweight cryptography)," CRYPTREC, Japan, Tech. Rep., Mar. 2017. [Online]. Available: https://www.cryptrec.go.jp/en/tech_guidelines.html

[43] *Lightweight Cryptography|CSRC*. Accessed: Oct. 30, 2020. [Online]. Available: https://csrc.nist.gov/projects/lightweight-cryptography

[44] A. Juels and S. A. Weis, "Authenticating pervasive devices with human protocols," in *Proc. 25th Annu. Int. Cryptol. Conf.* Berlin, Germany: Springer, Aug. 2005, pp. 293–308. [Online]. Available: https://link.springer.com/chapter/10.1007/11535218_18

[45] W. J. Okello, Q. Liu, F. A. Siddiqui, and C. Zhang, "A survey of the current state of lightweight cryptography for the Internet of Things," in *Proc. Int. Conf. Comput., Inf. Telecommun. Syst. (CITS)*, Jul. 2017, pp. 292–296.

[46] W. Stallings. (2017). *Cryptography and Network Security: Principles and Practice*. [Online]. Available: https://www.pearson.com/us/higher-education/product/Stallings-Cryptogra%phy-and-Network-Security-Principles-and-Practice-6th-Edition/9780133354690.htm%l

[47] T. Suzaki and K. Minematsu, "Improving the generalized feistel," in *Proc. 17th Int. Workshop Fast Softw. Encryption*. Berlin, Germany: Springer, Feb. 2010, pp. 19–39. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-13858-4_2

[48] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, T. Akishita, and F. Regazzoni, "Midori: A block cipher for low energy," in *Proc. 21st Int. Conf. Theory Appl. Cryptol. Inf. Secur., Part II*. Berlin, Germany: Springer, Nov./Dec. 2015, pp. 411–436. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-662-48800-3_17

[49] N. Pub, "197: Advanced encryption standard (AES)," *Federal Inf. Process. Standards*, vol. 197, no. 441, p. 0311, 2001.

[50] A. Moradi, A. Poschmann, S. Ling, C. Paar, and H. Wang, "Pushing the limits: A very compact and a threshold implementation of AES," in *Proc. 30th Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, May 2011, pp. 69–88. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-20465-4_6

[51] A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin, and C. Vikkelsoe, "Present: An ultra-lightweight block cipher," in *Proc. 9th Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, Sep. 2007, pp. 450–466. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-74735-2_31

[52] C. Rolfes, A. Poschmann, G. Leander, and C. Paar, "Ultra-lightweight implementations for smart devices-security for 1000 gate equivalents," in *Proc. 8th IFIP WG 8.8/11.2 Int. Conf. Smart Card Res. Adv. Appl.* Berlin, Germany: Springer, Sep. 2008, pp. 89–103. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-85893-5_7

[53] W. Zhang, Z. Bao, D. Lin, V. Rijmen, B. Yang, and I. Verbauwhede, "RECTANGLE: A bit-slice lightweight block cipher suitable for multiple platforms," *Sci. China Inf. Sci.*, vol. 58, no. 12, pp. 1–15, Dec. 2015.

[54] S. Banik, S. K. Pandey, T. Peyrin, Y. Sasaki, S. Sim, and Y. Todo. (2007). *Gift: A Small Present Towards Reaching the Limit of Lightweight Encryption (Full Version)*. [Online]. Available: https://infoscience.epfl.ch/record

[55] S. Banik, A. Chakraborti, T. Iwata, K. Minematsu, M. Nandi, T. Peyrin, Y. Sasaki, S. M. Sim, and Y. Todo, "GIFT-COFB," *Submission Round*, vol. 1, p. 29, Mar. 2019.

[56] Y. Liu and Y. Sasaki, "Related-key boomerang attacks on gift with automated trail search including bct effect," in *Proc. 24th Australas. Conf. Inf. Secur. Privacy (ACISP)*. Cham, Switzerland: Springer, Jul. 2019, pp. 555–572. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-21548-4_30

[57] C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, "FPGA-based assessment of Midori and GIFT lightweight block ciphers," in *Proc. 20th Int. Conf. Inf. Commun. Secur. (ICICS)* Cham, Switzerland: Springer, Oct. 2018, pp. 745–755. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-030-01950-1_45

[58] A. Adomnicai, Z. Najm, and T. Peyrin, "Fixslicing: A new gift representation," in *Proc. IACR*, 2020, p. 412.

[59] C. Beierle, J. Jean, S. Kölbl, G. Leander, A. Moradi, T. Peyrin, Y. Sasaki, P. Sasdrich, and S. M. Sim, "The skinny family of block ciphers and its low-latency variant mantis," in *Proc. 36th Annu. Int. Cryptol. Conf., Part II*. Berlin, Germany: Springer, Aug. 2016, pp. 123–153. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-662-53008-5_5

[60] T. Suzaki, K. Minematsu, S. Morioka, and E. Kobayashi, "Twine: A lightweight, versatile block cipher," in *Proc. ECRYPT Workshop Lightw. Cryptogr.*, 2011, pp. 1–5.

[61] S. Banik, A. Bogdanov, T. Isobe, K. Shibutani, H. Hiwatari, and T. Akishita, "Regaz Zoni, F.: Midori: A block cipher for low energy (extended version)," Cryptol. ePrint Arch., Tech. Rep. 2015/1142, 2015. [Online]. Available: https://eprint.iacr.org/2015/1142.pdf

[62] C. H. Lim and T. Korkishko, "mCrypton—A lightweight block cipher for security of low-cost RFID tags and sensors," in *Proc. Int. Workshop Inf. Secur. Appl.* Berlin, Germany: Springer, 2005, pp. 243–258.

[63] C. H. Lim, "A revised version of crypton: Crypton V1. 0," in *Proc. Int. Workshop Fast Softw. Encryption*. Berlin, Germany: Springer, 1999, pp. 31–45.

[64] J. Daemen, M. Peeters, G. Assche, and V. Rijmen, "The Noekeon block cipher," in *Proc. 1st Open NESSIE Workshop*, 2000, pp. 1–5.

[65] L. Knudsen and H. Raddum, "On Noekeon. Public reports of the Nessie project. Report: NES," New Eur. Schemes Signatures, Integrity Encryption (NESSIE Project), NES Rep. DOC/UIB/WP3/009/1, 2001. [Online]. Available: https://www.cosic.esat.kuleuven.be/nessie/reports/phase1/uibwp3-009.pdf

[66] F.-X. Standaert, G. Piret, G. Rouvroy, J.-J. Quisquater, and J.-D. Legat, "ICEBERG: An involutional cipher efficient for block encryption in reconfigurable hardware," in *Proc. Int. Workshop Fast Softw. Encryption*. Berlin, Germany: Springer, 2004, pp. 279–298.

[67] H. Cheng and H. M. Heys, "Compact ASIC implementation of the ICEBERG block cipher with concurrent error detection," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2008, pp. 2921–2924.

[68] C. Wang and H. M. Heys, "An ultra compact block cipher for serialized architecture implementations," in *Proc. Can. Conf. Electr. Comput. Eng.*, May 2009, pp. 1085–1090.

[69] H. Cheng, H. M. Heys, and C. Wang, "PUFFIN: A novel compact block cipher targeted to embedded digital systems," in *Proc. 11th EUROMICRO Conf. Digit. Syst. Design Archit., Methods Tools*, 2008, pp. 383–390.

[70] M. R. Albrecht, B. Driessen, E. B. Kavun, G. Leander, C. Paar, and T. Yalçın, "Block ciphers–focus on the linear layer (feat. pride)," in *Proc. Annu. Cryptol. Conf.* Cham, Switzerland: Springer, 2014, pp. 57–76.

[71] J. Borgho *et al.*, "PRINCE—A low-latency block cipher for pervasive computing applications," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur. (ASIACRYPT), Adv. Cryptol.* Springer, 2012, pp. 208–225. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-34961-4_14

[72] L. Batina, A. Das, B. Ege, E. B. Kavun, N. Mentens, C. Paar, I. Verbauwhede, and T. and Yalçın, "Dietary recommendations for lightweight block ciphers: Power, energy and area analysis of recently developed architectures," in *Proc. Int. Workshop Radio Freq. Identificat., Secur. Privacy Issues*. Berlin, Germany: Springer, 2013, pp. 103–112.

[73] L. Knudsen, G. Leander, A. Poschmann, and M. Robshaw, "Printcipher: A block cipher for IC-printing," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst. (CHES)*. Springer, 2010, pp. 16–32. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-642-15031-9_2

[74] Z. Gong, S. Nikova, and Y. W. Law, "KLEIN: A new family of lightweight block ciphers," in *Proc. Int. Workshop Radio Freq. Identificat., Secur. Privacy Issues*. Berlin, Germany: Springer, 2011, pp. 1–18.

[75] J. Guo, T. Peyrin, A. Poschmann, and M. Robshaw, "The LED block cipher," in *Proc. Int. workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2011, pp. 326–341.

[76] J. Guo, T. Peyrin, and A. Poschmann, "The PHOTON family of lightweight hash functions," in *Proc. Annu. Cryptol. Conf.* Berlin, Germany: Springer, 2011, pp. 222–239.

[77] E. Biham, "New types of cryptanalytic attacks using related keys," *J. Cryptol.*, vol. 7, no. 4, pp. 229–246, Dec. 1994.

[78] G. Piret, T. Roche, and C. Carlet, "Picaro—A block cipher allowing efficient higher-order side-channel resistance," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Berlin, Germany: Springer, 2012, pp. 311–328.

[79] B. Gérard, V. Grosso, M. Naya-Plasencia, and F.-X. Standaert, "Block ciphers that are easier to mask: How far can we go?" in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2013, pp. 383–399.

[80] H. Yap, K. Khoo, A. Poschmann, and M. Henricksen, "EPCBC—A block cipher suitable for electronic product code encryption," in *Proc. Int. Conf. Cryptol. Netw. Secur.* Berlin, Germany: Springer, 2011, pp. 76–97.

[81] M. R. Z'aba, N. Jamil, M. E. Rusli, M. Z. Jamaludin, and A. A. M. Yasir, "I-PRESENT: An involutive lightweight block cipher," *J. Inf. Secur.*, vol. 2014, p. 25, Jul. 2014.

[82] A. Poschmann, G. Leander, K. Schramm, and C. Paar, "New light-weight crypto algorithms for RFID," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2007, pp. 1843–1846.

[83] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann, and L. Uhsadel, "A survey of lightweight-cryptography implementations," *IEEE Des. Test. Comput.*, vol. 24, no. 6, pp. 522–533, Nov. 2007.

[84] P. Kumarkushwaha, M. P. Singh, and P. Kumar, "A survey on lightweight block ciphers," *Int. J. Comput. Appl.*, vol. 96, no. 17, pp. 1–7, Jun. 2014.

[85] M. Appel, A. Bossert, S. Cooper, T. Kußmaul, J. Löffler, C. Pauer, and A. Wiesmaier, "Block ciphers for the IoT-SIMON, SPECK, KATAN, LED, TEA, PRESENT, and SEA compared," Tech. Univ. Darmstadt, Darmstadt, Germany, Tech. Rep., 2016. [Online]. Available: http://download.mmag.hrz.tu-darmstadt.de/media/FB20/Dekanat/Publikationen/CDC/2016-09-05_TR_SimonSpeckKatanLedTeaPresentSea.pdf

[86] B. Andrews, S. Chapman, and S. Dearstyne, "Tiny encryption algorithm (TEA) cryptography 4005.705. 01 graduate team ACD final report," Rochester Inst. Technol., Rochester, NY, USA, Tech. Rep. 33695183, 2020. [Online]. Available: https://www.coursehero.com/file/33695183/TEApdf/

[87] P. Israsena and S. Wongnamkum, "Hardware implementation of a TEA-based lightweight encryption for RFID security," in *RFID Security*. Boston, MA, USA: Springer, 2008, pp. 417–433.

[88] D. Williams, "The tiny encryption algorithm (TEA)," *Netw. Secur.*, vol. 26, pp. 1–14, Apr. 2008.

[89] G. Sekar, N. Mouha, V. Velichkov, and B. Preneel, "Meet-in-the-middle attacks on reduced-round XTEA," in *Proc. Cryptograph. Track RSA Conf.* Berlin, Germany: Springer, 2011, pp. 250–267.

[90] J.-P. Kaps, "Chai-tea, cryptographic hardware implementations of XTEA," in *Proc. Int. Conf. Cryptol. India*. Berlin, Germany: Springer, 2008, pp. 363–375.

[91] J. Lu, "Related-key rectangle attack on 36 rounds of the XTEA block cipher," *Int. J. Inf. Secur.*, vol. 8, no. 1, pp. 1–11, Feb. 2009.

[92] D. J. Wheeler and R. M. Needham, *Correction to XTEA*. Cambridge, U.K.: Cambridge Univ. Press, 1998.

[93] K. Aoki, T. Ichikawa, M. Kanda, M. Matsui, S. Moriai, J. Nakajima, and T. Tokita, "Camellia: A 128-bit block cipher suitable for multiple platforms—Design and analysis," in *Proc. Int. Workshop Sel. Areas Cryptogr.* Berlin, Germany: Springer, 2000, pp. 39–56.

[94] A. Satoh and S. Morioka, "Hardware-focused performance comparison for the standard block ciphers aes, camellia, and triple-des," in *Proc. Int. Conf. Inf. Secur.* Berlin, Germany: Springer, 2003, pp. 252–266.

[95] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The simon and speck families of lightweight block ciphers," *IACR Cryptol. ePrint Arch.*, vol. 2013, no. 1, pp. 404–449, 2013.

[96] F.-X. Standaert, G. Piret, N. Gershenfeld, and J.-J. Quisquater, "SEA: A scalable encryption algorithm for small embedded applications," in *Proc. Int. Conf. Smart Card Res. Adv. Appl.* Berlin, Germany: Springer, 2006, pp. 222–236.

[97] F. Mace and F. Standaert, "ASIC implementations of the block cipher sea for constrained applications," in *Proc. 3rd Int. Conf. RFID Secur.*, 2007, pp. 103–114.

[98] T. Eisenbarth, Z. Gong, T. Güneysu, and S. Heyse, "Compact implementation and performance evaluation of block ciphers in attiny devices," in *Proc. Int. Conf. Cryptol. Afr.* Berlin, Germany: Springer, 2012, pp. 172–187.

[99] C. Rizzo and C. Brookson, *Security for ICT-the Work of ETSI*. Sophia Antipolis, France: ETSI, 2009.

[100] A. Satoh and S. Morioka, "Small and high-speed hardware architectures for the 3GPP standard cipher KASUMI," in *Proc. Int. Conf. Inf. Secur.* Berlin, Germany: Springer, 2002, pp. 48–62.

[101] M. Izadi, B. Sadeghiyan, S. S. Sadeghian, and H. A. Khanooki, "MIBS: A new lightweight block cipher," in *Proc. Int. Conf. Cryptol. Netw. Secur.* Berlin, Germany: Springer, 2009, pp. 334–348.

[102] W. Wu and L. Zhang, "LBlock: A lightweight block cipher," in *Proc. Int. Conf. Appl. Cryptogr. Netw. Secur.* Berlin, Germany: Springer, 2011, pp. 327–344.

[103] A. Poschmann, S. Ling, and H. Wang, "256 bit standardized crypto for 650 GE–GOST revisited," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2010, pp. 219–233.

[104] F. Karakoç, H. Demirci, and A. E. Harmancı "ITUbee: A software oriented lightweight block cipher," in *Proc. Int. Workshop Lightw. Cryptogr. Secur. Privacy*. Berlin, Germany: Springer, 2013, pp. 16–27.

[105] M. Kumar, S. K. Pal, and A. Panigrahi, "Few: A lightweight block cipher," *Turkish J. Math. Comput. Sci.*, vol. 11, no. 2, pp. 58–73, 2014.

[106] T. Shirai, K. Shibutani, T. Akishita, S. Moriai, and T. Iwata, "The 128-bit blockcipher CLEFIA (extended abstract)," in *Fast Software Encryption (FSE)* (Lecture Notes in Computer Science), vol. 4593. Springer, 2007. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-540-74619-5_12

[107] T. Akishita and H. Hiwatari, "Very compact hardware implementations of the blockcipher CLEFIA," in *Proc. Int. Workshop Sel. Areas Cryptography*. Berlin, Germany: Springer, 2011, pp. 278–292.

[108] C. Tezcan, "The improbable differential attack: Cryptanalysis of reduced round CLEFIA," in *Proc. Int. Conf. Cryptol. India*. Berlin, Germany: Springer, 2010, pp. 197–209.

[109] J. Hosseinzadeh and M. Hosseinzadeh, "A comprehensive survey on evaluation of lightweight symmetric ciphers: Hardware and software implementation," *Adv. Comput. Sci., Int. J.*, vol. 5, no. 4, pp. 31–41, 2016.

[110] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita, and T. Shirai, "Piccolo: An ultra-lightweight blockcipher," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2011, pp. 342–357.

[111] S. K. Ojha, "TWIS—A lightweight block cipher," in *Proc. Int. Conf. Inf. Syst. Secur.* Berlin, Germany: Springer, 2009, pp. 280–291.

[112] B. Su, W. Wu, L. Zhang, and Y. Li, "Full-round differential attack on TWIS block cipher," in *Proc. Int. Workshop Inf. Secur. Appl.* Berlin, Germany: Springer, 2010, pp. 234–242.

[113] S. S. M. AlDabbagh, I. F. T. Al Shaikhli, and M. A. Alahmad, "HISEC: A new lightweight block cipher algorithm," in *Proc. 7th Int. Conf. Secur. Inf. Netw.*, 2014, pp. 151–156.

[114] X. Lai and J. Massey, "A proposal for a new block encryption standard," in *Proc. Workshop Theory Appl. Cryptograph. Techn.*, 1991, pp. 389–404

[115] O. Tigli, "Area efficient ASIC implementation of IDEA (international data encryption standard)," *Best Des. ASIC Implement. IDEA, GMU*, 2003.

[116] S. Mukherjee and B. Sahoo, "A survey on hardware implementation of IDEA cryptosystem," *Inf. Secur. J., Global Perspective*, vol. 20, nos. 4–5, pp. 210–218, Jan. 2011.

[117] D. Hong, J. Sung, S. Hong, J. Lim, S. Lee, and B. Koo, "Hight: A new block cipher suitable for low-resource device," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2006, pp. 46–59.

[118] Y.-I. Lim, J.-H. Lee, Y. You, and K.-R. Cho, "Implementation of HIGHT cryptic circuit for RFID tag," *IEICE Electron. Exp.*, vol. 6, no. 4, pp. 180–186, 2009.

[119] J. John, "BEST-1: A light weight block cipher," *IOSR J. Comput. Eng.*, vol. 16, no. 2, pp. 91–95, 2014.

[120] D. Hong, J.-K. Lee, D.-C. Kim, D. Kwon, K. H. Ryu, and D.-G. Lee, "LEA: A 128-bit block cipher for fast encryption on common processors," in *Proc. Int. Workshop Inf. Secur. Appl.* Cham, Switzerland: Springer, 2013, pp. 3–27.

[121] D. Lee, D.-C. Kim, D. Kwon, and H. Kim, "Efficient hardware implementation of the lightweight block encryption algorithm LEA," *Sensors*, vol. 14, no. 1, pp. 975–994, Jan. 2014.

[122] N. Courtois, G. V. Bard, and D. A. Wagner, "Algebraic and slide attacks on KeeLoq," in *Proc. IACR*, 2007, p. 62.

[123] A. Bogdanov, "Linear slide attacks on the KeeLoq block cipher," in *Proc. Int. Conf. Inf. Secur. Cryptol.* Berlin, Germany: Springer, 2007, pp. 66–80.

[124] C. De Canniere, O. Dunkelman, and M. Knežević, "KATAN and KTANTAN—A family of small and efficient hardware-oriented block ciphers," in *Proc. Int. Workshop Cryptograph. Hardw. Embedded Syst.* Berlin, Germany: Springer, 2009, pp. 272–288.

[125] S. Das, "Halka: A lightweight, software friendly block cipher using ultra-lightweight 8-bit s-box," in *Proc. IACR*, 2014, p. 1104.

[126] D. Engels, X. Fan, G. Gong, H. Hu, and E. M. Smith, "Hummingbird: Ultra-lightweight cryptography for resource-constrained devices," in *Proc. Int. Conf. Financial Cryptogr. Data Secur.* Berlin, Germany: Springer, 2010, pp. 3–18.

[127] M.-J. O. Saarinen, "Cryptanalysis of hummingbird-1," in *Proc. Int. Workshop Fast Softw. Encryption*. Berlin, Germany: Springer, 2011, pp. 328–341.

[128] D. Engels, M.-J. O. Saarinen, P. Schweitzer, and E. M. Smith, "The hummingbird-2 lightweight authenticated encryption algorithm," in *Proc. Int. Workshop Radio Freq. Identificat., Secur. Privacy Issues*. Berlin, Germany: Springer, 2011, pp. 19–31.

[129] D. Dinu, A. Biryukov, J. Großschädl, D. Khovratovich, Y. Le Corre, and L. Perrin, "Felics–fair evaluation of lightweight cryptographic systems," in *Proc. NIST Workshop Light. Cryptogr.*, 2015, p. 128.

[130] D. Dinu, Y. L. Corre, D. Khovratovich, L. Perrin, J. Großschädl, and A. Biryukov, "Triathlon of lightweight block ciphers for the Internet of Things," *J. Cryptograph. Eng.*, vol. 9, no. 3, pp. 283–302, Sep. 2019.

[131] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers, "The SIMON and SPECK lightweight block ciphers," in *Proc. 52nd Annu. Design Autom. Conf.*, Jun. 2015, pp. 1–6.

[132] *Differential Fault Analysis—Wikipedia*. Accessed: Oct. 10, 2020. [Online]. Available: https://en.wikipedia.org/wiki/Differential_fault_analysis

[133] J. Breier, X. Hou, and Y. Liu, "Fault attacks made easy: Differential fault analysis automation on assembly code," in *Proc. IACR Trans. Cryptograph. Hardw. Embedded Syst.*, May 2018, pp. 96–122.

[134] A. Bogdanov, D. Khovratovich, and C. Rechberger, "Biclique cryptanalysis of the full AES," in *Proc. Int. Conf. Theory Appl. Cryptol. Inf. Secur.* Berlin, Germany: Springer, 2011, pp. 344–371.

[135] F. Zhang, Y. Zhang, H. Jiang, X. Zhu, S. Bhasin, X. Zhao, Z. Liu, D. Gu, and K. Ren, "Persistent fault attack in practice," in *Proc. IACR Trans. Cryptograph. Hardw. Embedded Syst.*, Mar. 2020, pp. 172–195.

[136] K. K, I. Roy, C. Rebeiro, A. Hazra, and S. Bhunia, "FEDS: Comprehensive fault attack exploitability detection for software implementations of block ciphers," in *Proc. IACR Trans. Cryptograph. Hardw. Embedded Syst.*, Mar. 2020, pp. 272–299.

[137] S. Bhasin, J. Breier, X. Hou, D. Jap, R. Poussier, and S. M. Sim, "SITM: See-in-the-middle side-channel assisted middle round differential cryptanalysis on SPN block ciphers," in *Proc. IACR Trans. Cryptograph. Hardw. Embedded Syst.*, Nov. 2019, pp. 95–122.

[138] K. Jeong, Y. Lee, J. Sung, and S. Hong, "Improved differential fault analysis on PRESENT-80/128," *Int. J. Comput. Math.*, vol. 90, no. 12, pp. 2553–2563, Dec. 2013.

[139] C. Blondeau and K. Nyberg, "Links between truncated differential and multidimensional linear properties of block ciphers and underlying attack complexities," in *Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn.* Berlin, Germany: Springer, 2014, pp. 165–182.

[140] O. Özen, K. Varıcı, C. Tezcan, and C. C. Kocair, "Lightweight block ciphers revisited: Cryptanalysis of reduced round PRESENT and HIGHT," in *Proc. Australas. Conf. Inf. Secur. Privacy*. Berlin, Germany: Springer, 2009, pp. 90–107.

[141] M. Renauld and F.-X. Standaert, "Algebraic side-channel attacks," in *Proc. Int. Conf. Inf. Secur. Cryptol.* Berlin, Germany: Springer, 2009, pp. 393–410.

[142] L. Yang, M. Wang, and S. Qiao, "Side channel cube attack on PRESENT," in *Proc. Int. Conf. Cryptol. Netw. Secur.* Berlin, Germany: Springer, 2009, pp. 379–391.

[143] B. Zhu, X. Dong, and H. Yu, "Milp-based differential attack on round-reduced gift," in *Proc. Cryptograph. Track RSA Conf.* Cham, Switzerland: Springer, 2019, pp. 372–390.

[144] Y. Sasaki, "Integer linear programming for three-subset meet-in-the-middle attacks: Application to gift," in *Proc. Int. Workshop Secur.* Cham, Switzerland: Springer, 2018, pp. 227–243.

[145] M. Cao and W. Zhang, "Related-key differential cryptanalysis of the reduced-round block cipher gift," *IEEE Access*, vol. 7, pp. 175769–175778, 2019.

[146] B. Zhao, X. Dong, W. Meier, K. Jia, and G. Wang, "Generalized related-key rectangle attacks on block ciphers with linear key schedule: Applications to SKINNY and GIFT," *Des., Codes Cryptogr.*, vol. 13, pp. 1–24, Dec. 2020.

[147] L. Dalmasso, F. Bruguier, P. Benoit, and L. Torres, "Evaluation of SPN-based lightweight crypto-ciphers," *IEEE Access*, vol. 7, pp. 10559–10567, 2019.

[148] P. Zhang and W. Zhang, "Differential cryptanalysis on block cipher skinny with MILP program," *Secur. Commun. Netw.*, vol. 2018, pp. 1–11, Oct. 2018.

[149] J. Ge, Y. Xu, R. Liu, E. Si, N. Shang, and A. Wang, "Power attack and protected implementation on lightweight block cipher SKINNY," in *Proc. 13th Asia Joint Conf. Inf. Secur. (AsiaJCIS)*, Aug. 2018, pp. 69–74.

[150] B. Nallathambi and K. Palanivel, "Fault diagnosis architecture for SKINNY family of block ciphers," *Microprocessors Microsyst.*, vol. 77, Sep. 2020, Art. no. 103202.

[151] J. H. Park, "Security analysis of mCrypton proper to low-cost ubiquitous computing devices and applications," *Int. J. Commun. Syst.*, vol. 22, no. 8, pp. 959–969, Apr. 2009.

[152] Y. Sun, M. Wang, S. Jiang, and Q. Sun, "Differential cryptanalysis of reduced-round ICEBERG," in *Proc. Int. Conf. Cryptol. Afr.* Berlin, Germany: Springer, 2012, pp. 155–171.

[153] C. Blondeau and B. Gérard, "Differential cryptanalysis of puffin and puffin2," in *Proc. ECRYPT Workshop Lightw. Cryptogr.*, 2011, p. 1.

[154] G. Zhao, B. Sun, C. Li, and J. Su, "Truncated differential cryptanalysis of PRINCE," *Secur. Commun. Netw.*, vol. 8, no. 16, pp. 2875–2887, Nov. 2015.

[155] Y. Lee, K. Jeong, C. Lee, J. Sung, and S. Hong, "Related-key cryptanalysis on the full PRINTcipher suitable for IC-printing," *Int. J. Distrib. Sensor Netw.*, vol. 10, no. 1, Jan. 2014, Art. no. 389476.

[156] Z. Ahmadian, M. Salmasizadeh, and M. R. Aref, "Biclique cryptanalysis of the full round KLEIN block cipher," *IET Inf. Secur.*, vol. 9, no. 5, pp. 294–301, Sep. 2015.

[157] J.-P. Aumasson, M. Naya-Plasencia, and M.-J. O. Saarinen, "Practical attack on 8 rounds of the lightweight block cipher KLEIN," in *Proc. Int. Conf. Cryptol. India*. Berlin, Germany: Springer, 2011, pp. 134–145.

[158] M. Gruber and B. Selmke, "Differential fault attacks on klein," in *Proc. Int. Workshop Constructive Side-Channel Anal. Secure Design*. Springer, 2019, pp. 80–95.

[159] G. Zhao, B. Sun, R. Li, L. Cheng, and C. Li, "Differential fault analysis on LED using super-sbox," *IET Inf. Secur.*, vol. 9, no. 4, pp. 209–218, Jul. 2015.

[160] E. Yarrkov, "Cryptanalysis of XXTEA," in *Proc. IACR*, 2010, p. 254.

[161] H. Tupsamudre, S. Bisht, and D. Mukhopadhyay, "Differential fault analysis on the families of Simon and speck ciphers," in *Proc. Workshop Fault Diagnosis Tolerance Cryptogr.*, 2014, pp. 40–48.

[162] A. Bay, J. Nakahara, and S. Vaudenay, "Cryptanalysis of reduced-round MIBS block cipher," in *Proc. Int. Conf. Cryptol. Netw. Secur.* Berlin, Germany: Springer, 2010, pp. 1–19.

[163] Y. Wang, W. Wu, X. Yu, and L. Zhang, "Security on LBlock against biclique cryptanalysis," in *Proc. Int. Workshop Inf. Secur. Appl.* Berlin, Germany: Springer, 2012, pp. 1–14.

[164] H. Soleimany, "Self-similarity cryptanalysis of the block cipher ITUbee," *IET Inf. Secur.*, vol. 9, no. 3, pp. 179–184, May 2015.

[165] T. Isobe, "A single-key attack on the full GOST block cipher," in *Proc. Int. Workshop Fast Softw. Encryption*. Springer, 2011, pp. 290–305.

[166] N. T. Courtois, "An improved differential attack on full gost," in *The New Codebreakers*. Berlin, Germany: Springer, 2016, pp. 282–303.

[167] S. A. Azimi, Z. Ahmadian, J. Mohajeri, and M. R. Aref, "Impossible differential cryptanalysis of piccolo lightweight block cipher," in *Proc. 11th Int. ISC Conf. Inf. Secur. Cryptol.*, Sep. 2014, pp. 89–94.

[168] J. Song, K. Lee, and H. Lee, "Biclique cryptanalysis on lightweight block cipher: HIGHT and piccolo," *Int. J. Comput. Math.*, vol. 90, no. 12, pp. 2564–2580, Dec. 2013.

[169] J. Huang, S. Vaudenay, and X. Lai, "On the key schedule of lightweight block ciphers," in *Proc. Int. Conf. Cryptol. India*. Cham, Switzerland: Springer, 2014, pp. 124–142.

[170] B. Koo, D. Hong, and D. Kwon, "Related-key attack on the full HIGHT," in *Proc. Int. Conf. Inf. Secur. Cryptol.* Berlin, Germany: Springer, 2010, pp. 49–67.

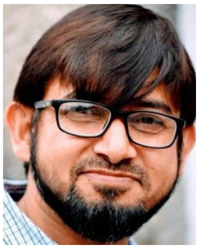
[171] A. Bogdanov, "Attacks on the KeeLoq block cipher and authentication systems," in *Proc. 3rd Conf. RFID Secur.*, 2007, pp. 1–5.

[172] B. Zhu and G. Gong, "Multidimensional meet-in-the-middle attack and its applications to KATAN32/48/64," *Cryptography Commun.*, vol. 6, no. 4, pp. 313–333, Dec. 2014.

[173] M.-J. O. Saarinen, "Related-key attacks against full Hummingbird-2," in *Proc. Int. Workshop Fast Softw. Encryption*. Berlin, Germany: Springer, 2013, pp. 467–482.

[174] (Mar. 2017). *Cryptographic Technology Guideline (Lightweight Cryptography)*. [Online]. Available: https://www.cryptrec.go.jp/report/cryptrec-gl-2003-2016en.pdf

VISHAL A. THAKOR is currently pursuing the Ph.D. degree in lightweight cryptography to improve security in resource-constrained IoT devices from Teesside University, U.K. He is also working as a part-time Lecturer with Teesside University. He is having more than ten years of teaching and around more than two years of industrial experience. Along with his passion to teach, he loves coding using C/C++, VB, C#, and Python. His areas of research interests include information security, cyber security, computer networks, algorithm design, data structure, the Internet of Things, and also Web designing. He is a lifetime member of Computer Society of India (CSI), India.



MOHAMMAD ABDUR RAZZAQUE (Member, IEEE) received the Ph.D. degree from UCD, Ireland. He worked as a Senior Research Fellow with the Trinity College Dublin, from October 2014 to January 2018, and a Senior Lecturer with UTM, Malaysia, from September 2011 to September 2014. He is currently a Senior Lecturer with the School of Computing, Engineering, and Digital Technologies, Teesside University. His research interests include centered on end-to-end IoT solutions and cybersecurity. He is an Editor of the *International Journal of Distributed Sensor Networks* and IoT JOURNAL.

MUHAMMAD R. A. KHANDAKER (Senior Member, IEEE) received the Ph.D. degree from Curtin University, Perth, WA, Australia. He worked as a Postdoctoral Research Fellow with University College London, U.K., in July 2013 to June 2018. He is currently an Assistant Professor with the School of Engineering and Physical Sciences, Heriot-Watt University. He is an Associate Editor of the IEEE WIRELESS COMMUNICATIONS LETTERS, IEEE COMMUNICATIONS LETTERS, and IEEE ACCESS.