Int. J. Nonlinear Anal. Appl. 13 (2022) 1, 3825-3851
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.6167

A systematic review of ultra-lightweight encryption algorithms

Noor Maher Naser[a], Jolan Rokan Naif[a,*]

[a]*Informatics Institute for Postgraduate Studies, Iraqi Commission for Computers and Informatics, Baghdad, Iraq*

(Communicated by Madjid Eshaghi Gordji)

Abstract

The Internet of Things (IoT) has opened a new era of technology and knowledge, several devices with limited resources are used. Those devices are vulnerable to a significant number of new malware and other emerging risks. Lightweight cryptographic algorithms are one of the most ideal approaches for safeguarding those IoT applications. Cryptography will conceal the data and by eliminating the possibility of obtaining any crucial information patterns, this assures that all data transmissions are secure, accurate, authenticated, permitted, and non-repudiable. Since using cryptographic algorithms for constrained or restricted devices is not ideal, the need for developing lightweight cryptographic algorithms increased through the last decades. Many lightweight blocks and stream ciphers are becoming common because they meet the requirements of low-power and constrained devices. This paper is a comprehensive survey in lightweight and ultra-lightweight encryption algorithms and includes both the recently proposed ciphers (Block and Stream), and the latest cryptanalysis results.

Keywords: Lightweight Cryptography, Ultra-Lightweight Cryptography, Block Ciphers, Stream Ciphers, Embedded Systems Security.

1. Introduction

Small computing devices including RFID tags, sensors, industrial controllers, and smart cards are becoming increasingly popular. The transition from desktop computers to mobile devices raises a slew of new security and privacy issues. Traditional cryptography protocols are difficult to implement on tiny devices. The tradeoff between security, performance, and resource needs in many traditional cryptographic protocols was tuned for desktop and server contexts, making them difficult

*Corresponding author
Email addresses: noor.sarsam@uoitc.edu.iq (Noor Maher Naser), newjolan@gmail.com (Jolan Rokan Naif)

Received: November 2021 *Accepted:* February 2022

or impossible to apply in resource-constrained devices. Their performance may not be satisfactory even if they are applied [49].

Lightweight cryptography is a subset of cryptography that focuses on creating solutions for low-resource devices. The academic community has put a lot of effort towards lightweight cryptography, such as efficient implementations of conventional encryption standards and the invention and analysis of novel lightweight algorithms and protocols [26]. In this paper a survey of Lightweight and ULTRA-Lightweight implementations of symmetric-key block and stream ciphers is presented.

The remainder of this paper is organized as follows. Section 2 provides an introduction to the conception of encryption algorithms, lightweight and ultra-lightweight encryption algorithms in general, symmetric-key block ciphers is overviewed in section 3, section 4 summarizes symmetric-key stream ciphers, a comparison of the lightweight encryption algorithms in section 5 and section 6 concludes this work.

2. Encryption Algorithms

People frequently use mobile devices and computers every day in a continuous manner and they might keep sensitive data on these personal devices (portable computers and mobile devices), they send these sensitive data via e-mail, instant messaging, and other forms of digital communications, when these personal devices are lost or stolen, or when a message is intercepted by a third party, that's when cryptography is necessary. Encryption software can protect this important information from being revealed by unwanted persons. since Encryption is the process of utilizing a software algorithm and a numeric key to turn readable data into an encoded form, as the key encrypts the data and converts it to ciphertext. The ciphertext contains all of the data in the plaintext, but it cannot be read by people or computers until it is decrypted. Therefore the process of turning ciphertext back to plaintext is known as Decryption, on the other hand Ciphers are the algorithms that are used to encrypt and decrypt data [53].

For securing sensitive and secret data stored on computers or sent over unsecured networks such as the Internet, a wide range of encryption software options are available, these tools encrypt data files and messages to keep them safe from unauthorized access, in case a computer is lost or stolen, or if a message is intercepted in transit [1].

The essential fields in security are: Availability, Authentication, Access Control, with Confidentiality, Non-Repudiation and Integrity as we may notice in Figure 1, all of these objectives can be achieved with the help of cryptographic primitives. Integrity and Confidentiality of the information are achieved by conventional cryptography since traditional cryptographic methods require large allocation of resources, correspondingly, IoT devices are identified as having limited computational power, limited power supply, limited memory and limited battery life. This surely defines why we need the development from conventional to lightweight cryptographic algorithms (LWC) to achieve these areas with the limitations of the IoT [50].

A. Lightweight Encryption Algorithms: Lightweight encryption is a type of encryption designed for devices with limited resources. The goal of a lightweight encryption is to provide security for devices with limited resources by using less memory, less computing resources, and less power or energy. The lightweight algorithms are also incredibly rapid in operation and handle small amounts of information, thanks to their exceptionally low resource and power consumption demands. Furthermore, it reduces the cost of implementing lightweight algorithms, making them more widely applicable [50].

An algorithm's weight is defined by the weight of a primitive which is roughly equal to the amount of time and space resources required for it to run. It can be measured in two different

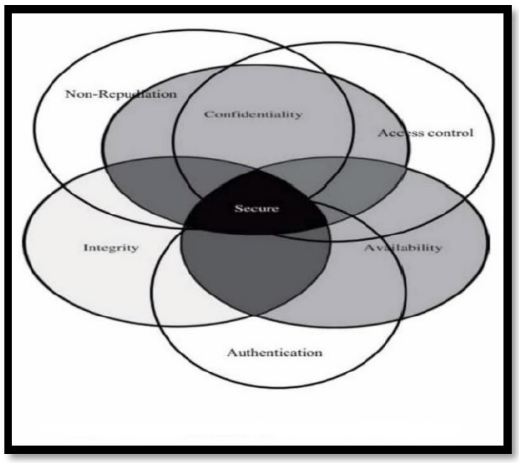

Figure 1: The main areas in the field of security [3]

status: in software and in hardware, the term lightweightness differs in both states, but what considered relevant is the power consumption, as you will notice we will only concentrate on software through this paper [66].

The time complexity of a primitive in software refers to two concepts, firstly, the algorithm's speed (which is determined by the number of clock cycles needed to process one byte of data) and secondly, the latency. While the space complexity is all about memory (RAM) plus the space required to store the algorithm [16].

Because of the restrictions of power transfer in RFID or because the gadget is powered by a battery, lightweight devices only have a limited amount of energy at their disposal. As a result, another important criteria is power usage, a low average power consumption in addition to a reasonable peak power consumption the device may have and surly both are measured in Watts as we may know [52].

Cryptography can be divided into two branches as we can see in figure 2 according to devices types

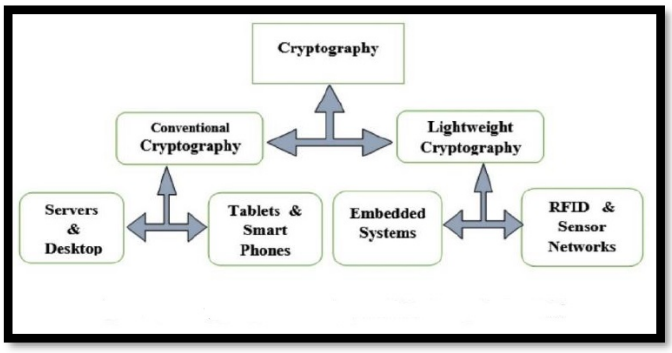

Figure 2: Cryptography Types according to devices

The lightweight algorithms, on the other hand, do not have a high bandwidth because they are meant to handle little amounts of data. Because of the limits of 1000 GE (Gate Equivalent) which is a unit of measurement for digital electronic circuits' manufacturing technology independent complexity. Light ciphers were largely intended for hardware implementation rather

than software. Quite furthermore, the constraints or limited resources for IoT devices make it difficult to adapt traditional algorithms to the needs of lightweight cryptography, Because of their limited processing resources and mathematical cost (complex mathematical operations), many of them have lost a lot of resistance. This, combined with cycles number in traditional cryptographic algorithms, leads to the loss of the memory and energy for limited resources devices, making traditional cipher algorithms unsuitable for IoT devices [50].

B. Ultra-Lightweight Encryption Algorithms: several metrics are considered when thinking about ultra-lightweight cryptography, these metrics are the benchmark for any cryptosystem and enhancing them will lead the encryption algorithm to be an ultra-lightweight (less GEs and less memory size) these metrics can be summarized as follows:

- Performance is achieved by measuring the execution time and the clock cycle number.
- The size of the used memory is achieved by measuring the consumption of RAM and ROM.
- The Security Level is achieved by measuring confusion and diffusion levels with the resistance to the multiple attacks (linear or differential) [67].

3. Lightweight Block ciphers

The block cipher is symmetric cipher uses an encryption technology that encrypts a block of n-bits of data of a defined size. Each cipher block is usually 64-, 128- or 256 bits in size. The majority of block cipher algorithms use iterated block ciphers to convert fixed-size plaintext blocks into identically sized ciphertext blocks. Block Cipher uses both confusion and diffusion principles, and the decryption in Block cipher is complex as compared to that of Stream Cipher [49].

Plaintext and ciphertext coherence in Block Ciphers should be as complicated as possible. As this was firstly specified by Claude Shannon (Theory of Secrecy Systems) published in 1949. He explained two important properties, Confusion and Diffusion, both should be fulfilled by a cipher. Confusion indicates that each character of the ciphertext should be dependent on several parts of the key, while Diffusion states that whenever one character in the plaintext is changed, multiple characters in the ciphertext should also change [30].

On the basis of internal structure, the block ciphers are mainly classified into two basic types: Substitution Permutation Networks (SPNs) and Feistel network [52], other references classify the block ciphers into five types: SPNs, Feistel networks, NLFSR-based, Hybrid and (ARX) Add Rotate-XOR [30].

many Feistel ciphers suffer from security problems unlike SPN ciphers, therefore in the field of lightweight cryptography SPN is more preferable to be a strong competitor than Feistel [18].

The next section shows the main lightweight and ultra-lightweight block ciphers with their characteristics, which will be summarized in section 5 in Table 1.

3.1. mCrypton

mCrypton presented by C.H. Lim and T. Korkishko in 2005 [44], it is considered a high-adapting algorithm because of the variable key length: 64 bit, 96 bit or 128 bit, with block size of 64 bit and 12 rounds. mCrypton divides every byte into two bits (two nibbles), It uses four 4×4 S-boxes because of its nonlinear transform components, each byte is divided into two 4-bit nibbles, as a result, all 16 nibbles were consumed. as shown in figure 3 there are four basic transformations in the mCrypton algorithm, starting with the nonlinear substitution and ending with the key addition [44, 52].

In 2019 a study proposed a mechanism based on an algorithm that combines mCrypton and Blowfish to solve the problem of security in data transmission [34].

Regarding security issue, The cipher was shown to be secure against all known block cipher attacks in the first security analysis. in [56], the authors noticed that mCrypton security is investigated in the related key settings. In [46], the authors construct 9-round related-key impossible differential attacks against mCrypton-96 and mCrypton-128. It was proved that mCrypton in the 8-round with 128-bit key is vulnerable to related-key rectangle attack.

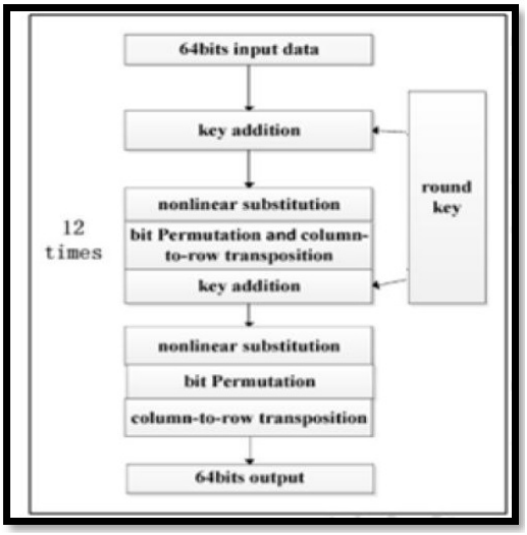

Figure 3: mCrypton algorithm encryption process [13]

3.2. PRESENT

PRESENT presented by A. Bogdanov et al. in 2007, It was one of the first ciphers to be used on devices with extremely little resources (ultra-constrained devices), it has an SPN structure, also it meets the requirements for lightweight and ultra-lightweight [14].

PRESENT is considered a big achievement in the development of L.W.Block Ciphers and for newer proposals it is used as a benchmark along with AES. It has been considered with CLEFIA a standards in ISO/IEC 29192 [38].

It uses a block size of 64-bit with a variable key size of 80 or 128 bit through 31 rounds, as shown in Figure 4, each round goes through one S-box and one P-Layer which is very simple and straight forward. The algorithm uses 4-bit input and output S-boxes in substitution layer for hardware optimization. It is one of the leanest lightweight algorithms (uses almost 1000 GE) and it beats other algorithms when it comes to area, security and performance but has a single drawback, which is the high-power consumption [14, 19].

Regarding security issue, several attacks have been reported like side-channel attacks, related-key attack and Biclique cryptanalysis on were reported in [10, 21], while in [24] proposed the differential fault analysis on the cipher. A truncated differential attack on the reduced 26-round cipher was presented in [13].

3.3. PUFFIN-2

PUFFIN-2 presented by Ch. Wang and H.M. Heys in 2009 and it is based on PUFFIN, after the late was broken due to statistical saturation attacks [20, 77].

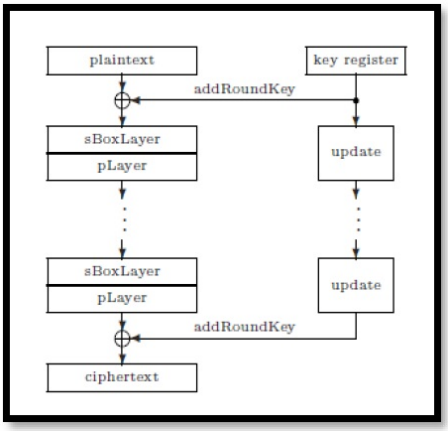

Figure 4: PRESENT Block Cipher Block Diagram [17]

It has serialized architecture SPN, with 64 bit block size and 80 bit key size through 34 rounds. Figure 5 shows the block diagrams of PUFFIN and PUFFIN-2.

Although it is fast and has small footprint (1083GE) for both encryption and decryption functionality; but its weakness is the Differential cryptanalysis. PUFFIN-2 consume extreme energy per bit resulting in low hardware efficiency. It produces high latency, low throughput, it also consumes low power. It is resistant to related-key attacks, since at key scheduling the relevant permutation layers are not regularly distributed among rounds. Differential cryptanalysis on the full cipher is slightly better than exhaustive search [12, 75].

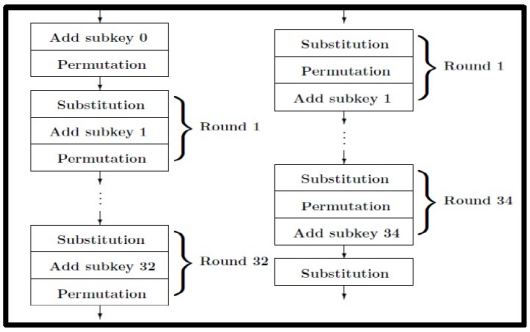

Figure 5: Puffin cipher (left) Block Diagram. Puffin-2 Cipher (right) Block Diagram [25]

3.4. SLIM

SLIM is presented by B. Aboushosha in 2020, the designers took into consideration power and area constraints while avoiding any compromise with security. As shown in figure 6, It is based on Feistel Structure, it has 32-bit Block size with 80-bit Key size through 32 rounds, by using 32 subkeys each of 16-bit which are generated from the 80-bit key. it also uses four (4×4) S-boxes, Despite the cipher's simple design and ease of implementation, it has a robust profile against the most malicious cryptanalyses. As a result of all these good characteristics, SLIM proved to be easily implemented with resource constrained devices such as RFID and mostly suitable for the internet of Health Things [1]. While designing SLIM, simplicity, security, accomplishing both confusion and diffusion concepts were taken into account.

The algorithm's fundamental feature is having a minimal footprint area suitable for RFID applications. The cryptosystem structure was created with the goal of being simple to implement in both

software and hardware. According to The algorithm SLIM is secured from linear and differential cryptanalysis [1, 24, 70].

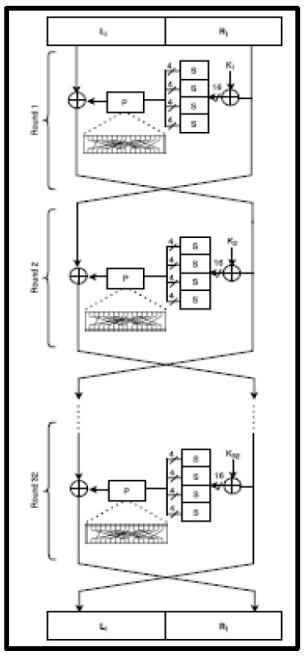

Figure 6: SLIM Block Cipher Encryption [30]

3.5. Klein

Klein presented by Z. Gong et al. in 2012, it has SPN structure that works on 64-bit Block size with variable key length of 64, 80 or 96-bit through 12, 16, and 20 rounds respectively. The round is made up of four layers in order: AddRoundKey, SubNibbles, RotateNibbles, and MixNibbles. The most used and known key is 64 bit key [30, 29].

The process can be explained easily in figure 7, for a state that enters a round, first it's Xored with the round-key through AddRoundKey. The outcome is then divided in 16 nibbles that are all transformed by the same (4×4) S-box. After that, RotateNibbles turns the state two bytes to the left of the 16 output nibbles of the S-boxes and finally MixNibbles applies Rijndael MixColumn transformation to each half of the state. The output of the final step will be the inputs for the next round encryption process [58].

Regarding security issue, several attacks on Klein were noticed through researches like practical attacks on Klein-64 [6], then an asymmetric biclique attack on the full version cipher was noticed above all The fact that the final layer (MixColumn) transformation does not correctly combine higher and lower nibbles could be the cipher's fundamental flaw [39].

3.6. SEA

SEA (Scalable Encryption Algorithm) presented by F.X. Standaert et al. in 2006, it was designed for scalable implementations on constrained devices with a focus on low-cost embedded settings [71]. The designers aimed that the cipher SEA will meet these requirements: low memory, small code size and limited instruction set above all that the efficient combination of encryption and decryption [30].

$SEA_{n,b}$ operates on various text, key and word sizes. As shown in figure 8, is based on a Feistel structure with a variable number of rounds, according to many reads SEA can be defined with respect to the following parameters:

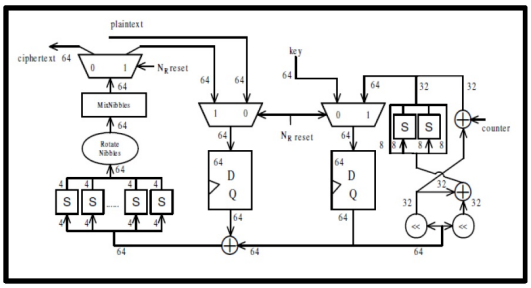

Figure 7: The data path of KLEIN-64 encryption [**?**]

- n is the plaintext size and the key size.
- B is the processor size.
- $nb = n/2b$ (is the number of words per Feistel branch).
- nr is the number of rounds.

While compared to other ciphers like PRESENT [37], the block size and the key-length must be in 6 bits and It can't be dependent on the bits of the processor. The mathematical operations in SEA are in order: XOR, rotations, 2n mod addition, and substitution.

Regarding security issue, the cipher is strong against linear and differential analysis attacks due to key size and variable number of operations. However, the number of rounds with the key size and the table of rules used all make the Cipher require longer time to implement [45].

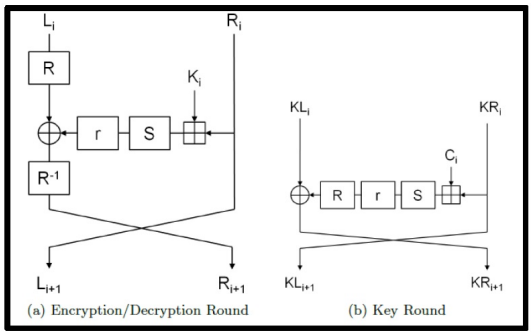

Figure 8: SEA Block Cipher Rounds [34]

3.7. CLEFIA

CLEFIA presented by T. Shirai et al. in 2007 [64], designed by SONY and is standardized with PRESENT in ISO/IEC 29192 [38], it is a lightweight block cipher identified for its extremely effective hardware and software implementations, it uses 128-bit of block size of and a variable key length of 128, 192 or 256-bit through 18, 22, and 26 rounds respectively, which is compatible with AES. The most lightweight encryption/decryption version uses 2604GE, It makes use of two types of 8-bit S-boxes [7, 30].

The designers of CLEFIA mentioned in [69] that CLEFIA consists of two parts:

- data processing part.
- key scheduling part.

Figure 9 shows that CLEFIA employs a generalized Feistel structure with four data lines, in addition to that there are key whitening parts at the beginning and the end of the cipher. By employing numerous new design strategies, CLEFIA delivers sufficient protection against known threats as well as flexibility for efficient implementation in both hardware and software [64].

The finest cryptanalysis results come from improbable differential attacks on reduced round versions, which are marginally better than exhaustive search [74]. in [41] the authors proposed in details an integral distinguisher of CLEFIA on a 9-round based on byte-pattern. And by using the partial sum technique the authors improved the previous result on 11- round CLEFIA and proposed integral attack on 12-, 13- and 14- round CLEFIA with the keys.

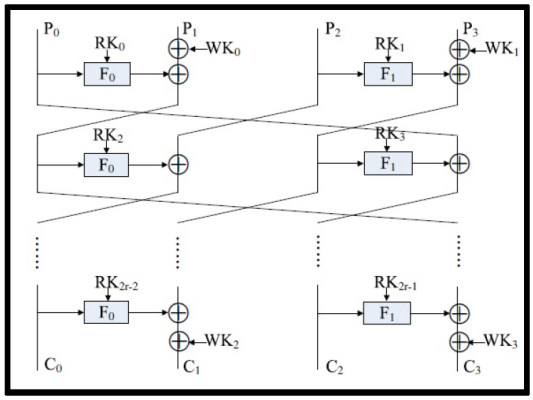

Figure 9: Clefia Block Cipher [36]

3.8. LBlock

LBlock presented by W. Wu and L. Zhang in 2011, it is an ultra-lightweight block cipher, was proposed at ACNS 2011, has a block size of 64-bit and key size of 80-bit through 32 rounds. In hardware, it occupies 1320GE, while it produces good software efficiency, it requires 3955 clock cycles to encrypt a single block [79].

The encryption technique is 4-bit focused, and is has a Feistel structure that has variation property, which means it can be implemented effectively in both hardware and software. Furthermore, the round function uses an SP-network, in which the confusion layer is built up of small 4×4 S-boxes and the diffusion layer is a simple 4-bit word permutation as shown in figure 10. All of these components were in the designer's consideration for achieving security and implementation efficiency [67].

The full 32 round LBlock achieves enough security margin against known attacks like differential cryptanalysis, linear cryptanalysis and Impossible Differential Cryptanalysis, but for integral attacks, the authors in [22] presented 24-round integral attacks on LBlock. It is the longest round that can be attacked by single-key attacks.

3.9. TWINE

TWINE presented in Japan scholar in SAC by T. Suzaki et al. in 2012, it is a lightweight block cipher, it has 64-bit block size along with variable key length of 80 and 128 bit and the no. of rounds is 36. the hardware implementation is 1866GE. It is a Generalized Feistel Network (GFN) and perform both operations of encryption and decryption integrally. Although many Comparable choices to LBlock have been made on Twine in order to make equilibrium performance on both states (hardware and software), Two of the most important differences between TWINE and LBlock is the number on S-boxes and the permutation process. TWINE uses a single 4-bit S-box and a 4-bit XOR

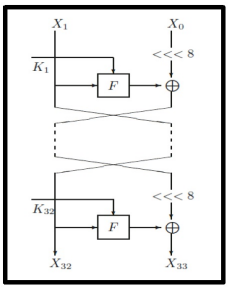

Figure 10: LBlock Block Cipher [39]

beside that it is completely bit permutation-free, [72] as shown in figure 11. The features of TWINE is limited with:

- no bit permutation.
- generalized Feistel based.
- no Galois-Field matrix.

Regarding security issues, Twine demonstrates that it offers a high level of protection, significantly against differential and saturation attacks [24]. While in the [82] authors proposed an electromagnetic analysis attack method on TWINE, there experimental results exposed TWINE's vulnerability when it is implemented as hardware.

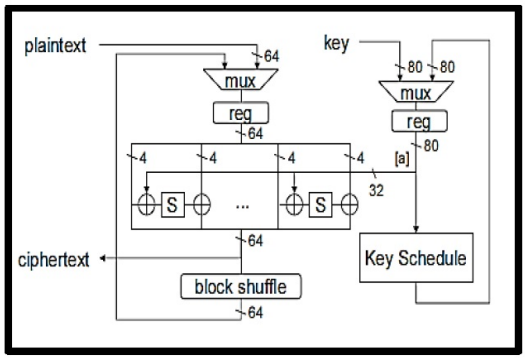

Figure 11: Twine Block Cipher [40]

3.10. μ^2

μ^2 is a lightweight block cipher introduced by Yeoh et al. in 2020, It was created with the goal of providing strong security while keeping limited resource devices performes properly, it is based on Generalized Feistel Structure (GFS) with a round function [81].

the name of the cipher was inspired by the authors from the micro (μ), which is often referred to a very little value and the power of two indicates the usage of two ciphers in one cipher.

The authors in [81] proved μ^2 to be more efficient comparing to PRESENT considering similar security issues, that's why it is consedered the most appropriate candidate for resource-constrained devices. It consists of 64-bit blocks with 80 bit of key-size through 15 rounds using only one S-Box, figure 12 shows the structure of the cipher.

Regarding security isses, the cipher is resistant to all well-known attacks [45].

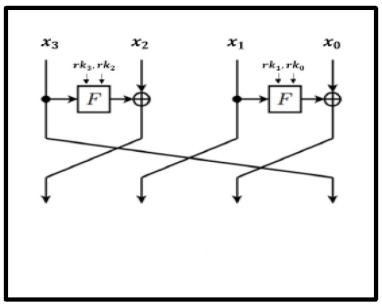

Figure 12: μ^2 Block Cipher Structure [43]

3.11. ANU

ANU presented by G. Bansod et al. in 2016, it is an ultra-lightweight block cipher, it is a balanced Feistel-based network, has 64-bit of block size with two variable keys 80 and 128-bits through 25 rounds as shown in figure 13, it has a small memory size and consumes very little power, and only needs less gate equivalents, it is best suited for very constrained applications like the applications of Internet of Things, it is considered a resource and energy efficient as well. The designers of ANU cipher took into consideration these good characteristics and will lead ANU to have a positive influence in the field of lightweight cryptography. The key-schedule of ANU is inspired by the key-schedule of PRESENT. the cipher has been designed to give optimal performance on both software and hardware platforms [8].

regarding security issue, the authors in [8] proved that ANU's design has a good resistance to almost all the known cryptanalysis.

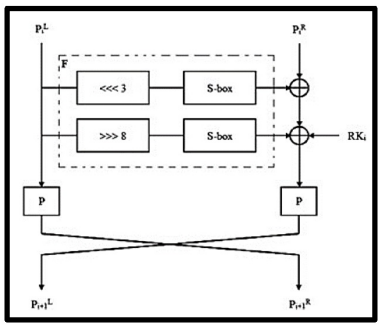

Figure 13: ANU Block Cipher [44]

3.12. ANU-II

ANU-II presented by V. Dahiphale et al in 2017, it is consedered an improved and modified version of the late ANU [8], It has a Feistel structure with block size of 64-bit supporting key length of 80 or 128-bits and the no. of rounds are 25 round [23].

ANU-II is resistant to basic and advanced attacks on lightweight ciphers due to the exsistance of P-layer. It is known for consuming very low power by using the least no. of rounds. As shown in figure 14 ANU-II is designed to have one S- Box, small no. of shift operators and two XOR gates, by that small design it is consedred the smallest lightweight cipher till date of publication relating to execution time, memory requirement and power consumption [23].

The cipher is proven to be secure against all well know attacks. By comparing the 2 versions, one can find that ANU-II surpasses ANU in throughput, execution time, and in requiring less power, it

even requires less power than PRESENT [14] which is consedered the most efficient block cipher in constrained environment.

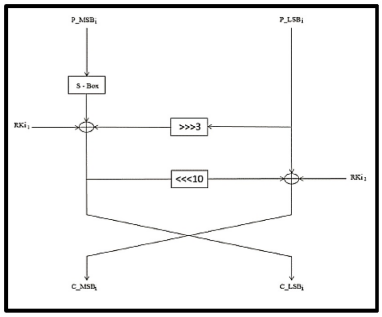

Figure 14: ANU-II Block Cipher [45]

3.13. NLBSIT

NLBSIT presented by Al-Ahdal et al., in 2020, it was designed as a new lightweight block cipher for resource restricted IoT computers, this cipher increases data security by integrating the benefits of both Feistel and SPN designs with an extra linear box idea. What makes NLBSIT a component for encryption in IoT devices is that it consumes less energy than other algorithms (encryption / decoding cycles) also consume less memory [3].

as shown in figure 15 it uses a key of 64-bit to encode 64-bit block size it also uses un-complicated mathematical operations with fewer rounds (XOR, XNOR, shifting and swapping). It has many advantages including less memory and less energy consumption and the algorithm has high speed, furthermore it has a high level of security due to the usage of both the SP and Feistel architectures in its creation, it thwarts all known attacks [3].

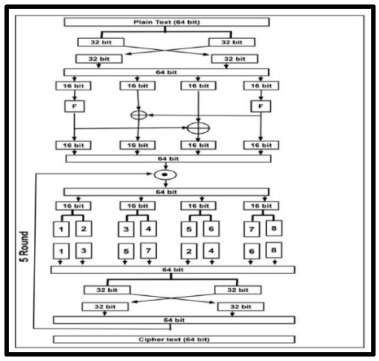

Figure 15: NLBSIT Block Cipher Encryption Process [46]

3.14. Piccolo

Piccolo the original version was presented by K. Shibutani et al. in 2011, while the novel architecture have been presented by Rahiyanath T.Y. in 2015 [59], it is an ultra-lightweight block cipher uses variant GFN as its structure with a permutation based key schedule [63] It supports 64 bit of block size with 2-key lengths of 80 and 128 bits through 25 and 31 rounds respectively. Piccolo has two versions depending on key length they are Piccolo-80 Piccolo-128, both consist of two parts:

- data processing part.

- key scheduling part.

It is implemented in a very compact manner and consumes very little energy. Piccolo-80 is the most lightweight according to [30], as the lightweightness is measured in GE. Figure 16 shows the architecture of Piccolo. The designers of Piccolo [63] took into consideration not to add flip-flops since it requires large area, likewise they used the same concept but by using logic gates , they used (AND-NOR) instead of XOR and used (OR-NAND) instead of XNOR for they take less area.

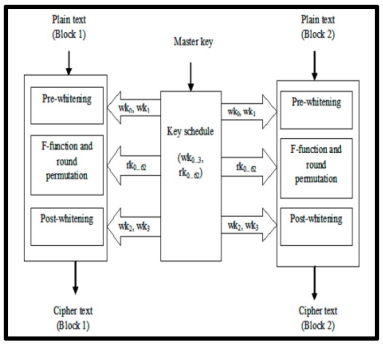

Figure 16: Piccolo Block Cipher Novel Architecture [47]

3.15. BORON

BORON proposed by Bansod et al. in 2017 [9], it is an ultra-lightweight block cipher that uses SPN, it operates on a 64-bit block and supports two key length of 80 and 128 bits and has a total of 25 rounds, it has a compact structure and low power, the design includes 3 important operations presented in figure 17 which are:

- Shift operators
- Round permutation layers
- XOR operations [9].

The cipher's unusual architecture allows it to produce a high number of active S-boxes in a small number of rounds, thwarting linear and differential attacks. BORON [30] shows good performance concerning hardware and software platforms, when compared to other existing SPN ciphers, BORON uses less power and has a higher throughput. Its compact design and low power dissipation make it ideal for applications with a small footprint.

Regarding security issue, the authors of [43] discussed and revealed that BORON shows a reasonable level of resistance against known attacks like linear, differential, algebraic, related key and slide attacks.

3.16. RECTANGLE

RECTANGLE proposed by Zhang et al in 2015, an ultra-lightweight block cipher, it has 64-bits of block size, a variable key of 80 or 128-bits iterating through 25 rounds based on SPN as shown in figure 18, it has a substitution layer consists of 16 (4×4) S-boxes connected in parallel and a permutation layer which compose of 3 rotations. It was proved to have great hardware and software performance and that will provides enough flexibility for different IoT application [83]. The 3 main advantages of RECTANGLE are:

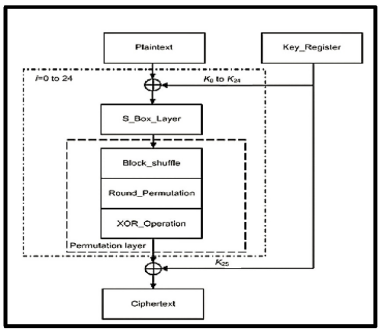

Figure 17: BORON Block Cipher Block Diagram [50]

- It is extremely hardware-friendly.
- It achieves a very competitive software speed among the existing lightweight block ciphers due to its bit-slice techniques (Very fast encryption speed).
- Because of the new design and selection of S-Box presented in [42] the asymmetric design of the permutation layer, RECTANGLE achieves a very good security-performance tradeoff.

However, the algorithm's non-robust round key generation appears to be its weakest point. Regarding security issues, the full round RECTANGLE is enough to resist some well-known attacks like linear, differential, SS attack, saturation attack and key schedule attacks [42, 83].

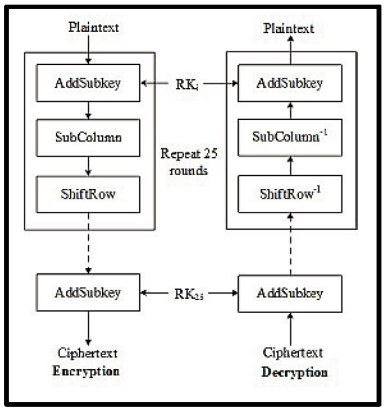

Figure 18: RECTANGLE Block Cipher Structure [52]

3.17. LICI

LICI proposed by Patil et al in 2017, it is a Feistel based ultra-lightweight block cipher that consists of 64 bit block size, 128 bit key length iterating through 31 rounds, the block diagram of LICI is shown in figure 19. LiCi's design performs flawlessly on both hardware and software platforms, as compared to the existing cipher it requires less footprint area, has less memory and consumes low power, this cipher is well suited for application where small footprint area and low power dissipation are important design metrics [57].

Regarding security issues, LiCi encryption algorithm has good resistance to the linear, differential attack, Biclique and Zero correlation and in [80] the authors discussed a new method to protect LiCi from side channel attack.

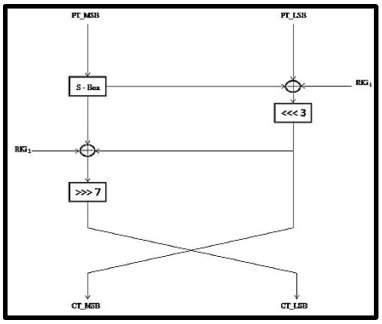

Figure 19: LiCi Block Cipher [54]

3.18. QTL

QTL proposed by Lang Li et al. in 2016, an ultra-lightweight block cipher suited for devices with a limited amount of resources, its structure is GFN, it has 64 bits of block size, variable key length of 64 or 128 bit iterating through 16, 20 rounds respectively and has a lot of S-boxes in the encryption process. Using SPNs diffusion in the design was to improve the security of a Feistel structure cipher, while not using a key schedule was intended to minimize hardware implementation's energy consumption [40]. Figure 20 represents the block diagram of QTL.

Regarding security issues, the authors of [40] have showed that the cipher offers a suitable security level against classic attacks. Especially, the differential and linear attacks. QTL achieves not only compact H.W implementation but also high security as authors in [65] claimed, in contrary the authors in [62] showed their disapproval opinion by experimenting the cipher and founding it susceptible to some attacks.

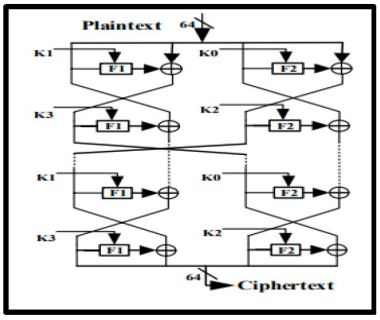

Figure 20: QTL Block Cipher [58]

4. Lightweight stream ciphers

Stream ciphers are symmetric ciphers that generate cipher text by encrypting plain text bit streams with associated key streams [51]. Encryption on Stream Ciphers is all about the conversion of plain text performed by taking one byte of the plaintext at a time (at most 8 bits could get converted at a time), and for decryption XORing the encryption text will easily reverse the plaintext. Lightweight stream ciphers are typically fast, compact, has maximum security and consume low power, making them an attractive choice for resource-constrained devices and resource constrained IoT environments, like low-power RFID tags. Stream ciphers have gained popularity in recent years as a result of various research into that field, they are useful in ubiquitous computing applications

and they can be used to secure the communication in applications where the plaintext length is either unknown or continuous, like network streams, mostly used in military applications [11].

Stream cipher also known as State Cipher, since encryption of each digit is dependent on the current state of the cipher. While evaluating both types, the stream cipher distinguishes as being more efficient in H.W applications, more rapid and easier. Stream ciphers are based on 2 constructions: Linear Feedback Shift Registers (LFSRs) and Non-Linear Feedback Shift Registers (NLFSRs), and it uses only confusion principle for the conversion [75].

Still, as pointed out by the eSTREAM [61] Stream Cipher Project, it is obvious that stream ciphers could continue to play an important role in constrained IoT application where high throughput remains critical and resources are very restricted.

The next section shows the main lightweight and ultra-lightweight stream ciphers with their characteristics, which will be summarized in section 5 in Table 2, based on their constructions, key size, and throughput.

4.1. LOGIC

LOGIC published by Ding et al. in 2019, it is a lightweight stream cipher that was generated by two Nonlinear Feadback Shift Registers (NFSRs) and a chaotic system. It is used in resource-constrained devices or constrained-environments [27].

The main idea of the cipher was combining a NFSR with a chaotic system, Since Chaotic systems have many good cryptography characteristics, using this idea will produces for the first time a new L.W stream cipher system based on chaos and it was named LOGIC, the algorithm is hardware-oriented. As shown in figure 21, LOGIC has a secret key of 80-bit divided on 2-NFSRs (40 bit each) with 3-multiplexers, a filter function and Logistic chaotic system.

Regarding security LOGIC stream cipher is considered good in resisting many fundamental crypt-analysis attacks due to the high nonlinearity and the good elasticity of the function used beside the existence of the two NFSRs [27].

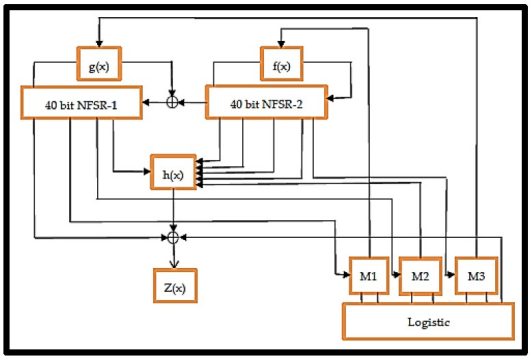

Figure 21: LOGIC Stream Cipher [65]

4.2. TRIVIUM

TRIVIUM presented by C.D. Cannière in 2006, It is an eSTREAM Profile 2 finalist and an ISO/IEC 29192-3: 2012 standardized stream cipher for LWC, it uses 80-bit key and 80-bit initialization vector (IV) and it consist of three interconnected non-linear feedback shift registers (NLFR) of length 84, 93 and 111 bits respectively, figure 22 shows the structure of TRIVIUM [17].

Its main idea was taken from the concepts of block ciphers, which was exchanging the blocks by equivalent stream cipher components. The only disadvantage of TRIVIUM is the use of so many flip flops for its hardware realization [76].

Regarding security issues, there has been some reports about several attack happened due to the simplicity of TRIVIUM, like cube attacks, improved differential fault analysis attacks, the authors of [54] proposed an attack and a low-cost countermeasure against the same attack on Trivium (a way to strike and defend against the same attack) [17, 76].

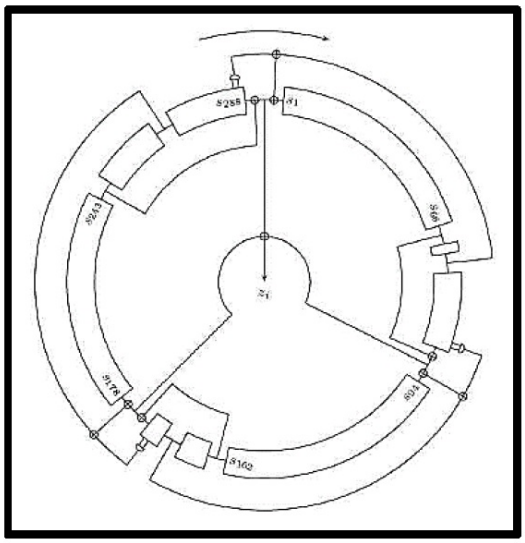

Figure 22: TRIVIUM Stream Cipher Structure [66]

4.3. A4

A4 proposed by N.A. Mohandas et al. in 2020, a lightweight stream cipher that uses in its construction one Linear Feedback Shift Register (LFSR) and One Feedback with Carry Shift Register (FCSR) with key size of 128 bit [51].

A4s algorithm has two parts:

- The sender side (Encryption Algorithm).
- The receiver side (Decryption Algorithm).

And between these two algorithms there are:

- A seed box (consisting of 256 values randomly ordered, each one is 128 bit length, set at both ends)
- An equation which will generate a pseudo-random value that will fit in 16×16 matrix [51],

$$(3 + CounterValue^2)MOD\ 255... \tag{4.1}$$

the value obtained will be converted to binary and passed to LFSR.

- Finally, the Counter Value, which will be used in the equation (4.1) [51, 60].

At both parts (sender and receiver ends), The LFER primarily clocks to set a number of times after receiving the seed value. This clocking of the LFSR ensures the second level security as an attacker is fully unaware of this computation, and this will secure the system. The FCSR, on the other hand, creates the key stream for encrypting and decrypting messages at the server and client,

respectively. A4 has a low time and space complexity compared to other algorithms because it only uses one LFSR and one FCSR.

Regarding security issues, Because of the arrangement of LFSR and FCSR, the cipher is completely resistant to algebraic attacks. This approach also stands up to brute-force and differential attacks.

4.4. Fruit-v2 and Fruit-80

- Fruit-v2 proposed by V.A. Ghafari et al. in 2016, a modified version of the original version Fruit which was broken by many attacks because of the low nonlinearity of its filter function. Its structure as shown in figure 23, consisted of LSFR, NSFR, 2 counters (7 and 8 bits respectively), the key size is 80 bit with 70 bit of IV expanding to 130 bits, from the idea of the key being reused by different IVs came the design of smaller area size ciphers (ultra-lightweight ciphers), hence Fruit-v2 is ultra-lightweight stream cipher [4].

  there have been several comparisons for Fruit with Grain since the later is also ultra-lightweight and considered the lightest candidate in the eSTREAM profile [25]. There has been studies on how resistant the main structure of Fruit since it was produced in 2016 against many old and new lightweight stream ciphers [26], and according to [73] it could be used as lightweight cryptographic protocol for smart homes, Relatively secure to Related key attacks, Cube attack, algebraic attack and Weak key IV.

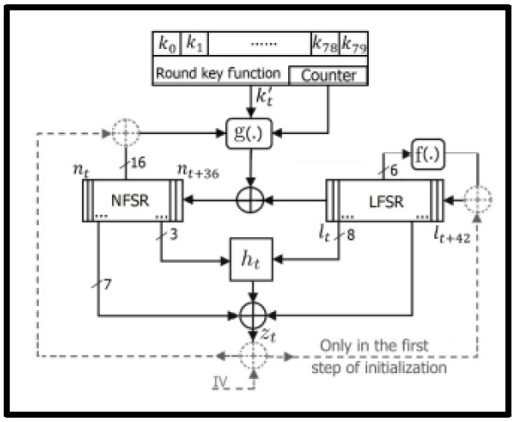

Figure 23: FRUIT-v2 Stream Cipher Block Diagram [71]

- Fruit-80 In 2018, V.A. Ghafari et al. proposed Fruit-80 as the final version of Fruit, to be much easier to implement and more secure than Fruit-v2, the size of LFSR and NFSR is only 80 bits (for 80-bit security level), while for resistance to the classical (TMDTO) attacks is higher, the internal state size should be at least twice that of the security level, to accomplish that the designers used new design ideas as shown in figure 24 [4].

  Fruit-80 is better than other small-state stream ciphers in terms of the initialization speed and area size [15].

4.5. Enocoro family

presented by D. Watanabe et al. in 2008 [78], it is a lightweight stream cipher family of PRNGs, it is standardized by IEC and ISO in ISO/IEC 29192-3, it has two versions named according to their key lengths as Enocoro-80 and Enocoro-128v2, its main advantage is low power consumption [35].

Enocoro is similar to Trivium in the length of the initialization phase, while unlike Grain concerning the encryption speed [47].

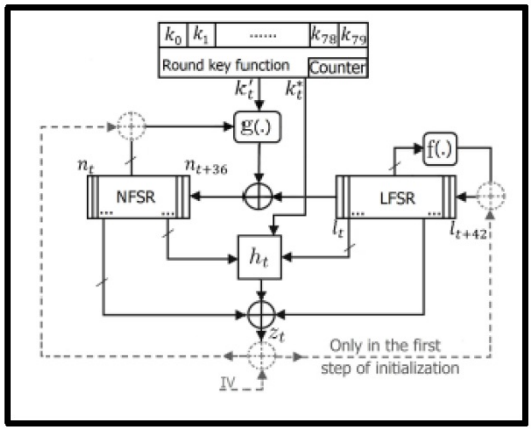

Figure 24: FRUIT80 Stream Cipher Block Diagram [75]

- Enocoro-80 presented to be comparable to eSTREAM profile 2 candidates, its software implementations achieve better performance than most of other candidates. Has a PRNG structure (pseudorandom number generator), a PRNG includes a finite state machine (FSM) and a linear feedback shift register (LFSR) is a class of FSMs, as shown in figure 25, whose update function being a linear map of canonical form, the cipher considered secure as there has not been any attacks ever violated the full-round security of the cipher [78].

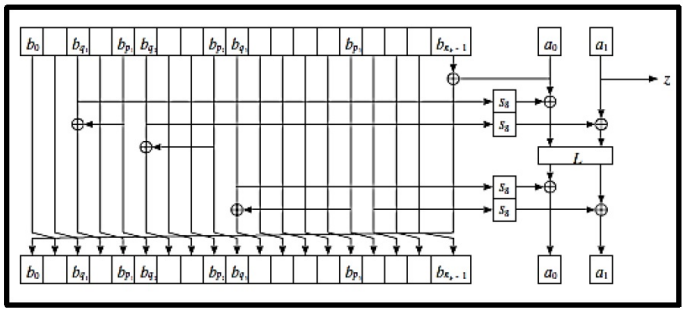

Figure 25: ENOCORO-80 Stream Cipher Structure [78]

- Encoro-128v2 presented after Enocoro-128v1-1 been discovered vulnerable against the related-key attack. Enocoro-128v2 is considered an update to the cipher to improve its security against such attacks [55]. it can be noted that Enocoro-128v2 and Enocoro-128v1.1, differ only in the choice of the characteristic polynomial over the function and in the way initialization is done [31]. It has a 128-bit key and a 64-bit IV value. Along with this, it is included in ISO/IEC 29192 International Standard for a lightweight stream cipher method (ISO/IEC 29192-3:2012), figure 26 shows the structure of this cipher.

4.6. The Grain Family

- The first cipher of the Grain family is called version 0, it is an unpublished version of the cipher Grain but was submitted to the eStream project, since there was many successful cryptanalyses attempts made by independent researchers, therefor the designers of the cipher found that using internal function was not feasible , and since it has security issues there was a simple modification which is a simple change in the output function also a minor change in the feedback function in the NFSR, due to some cryptanalysis suggestions produces Grain version 1 [33].

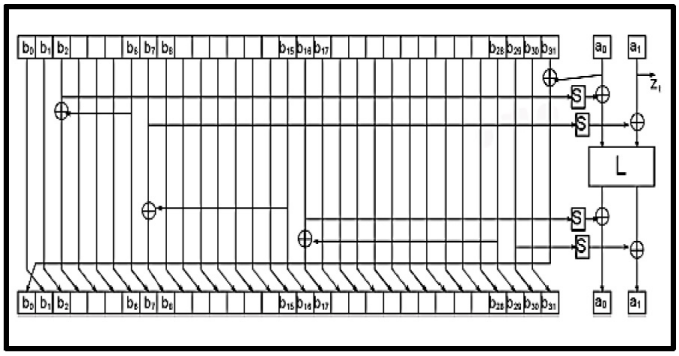

Figure 26: ENOCORO-128v2 Stream Cipher structure [16]

- Grain proposed by M. Hell et al. at 2007 and was submitted to the eSTREAM project, a stream cipher that targets hardware environments where gate count, power consumption and memory is very limited. As shown in figure 27 It is based on two shift registers (LFSR and NFSR) and a nonlinear filter function. The cipher also has the advantage of being able to enhance its performance without adding new hardware, both registers are 80 bit in size, the key size is 80 bits and the size of IV is 64 bit, Other hardware-oriented stream ciphers like E0 and A5/1, which are considered not secure, compare favorably in terms of hardware complexity and throughput [33].

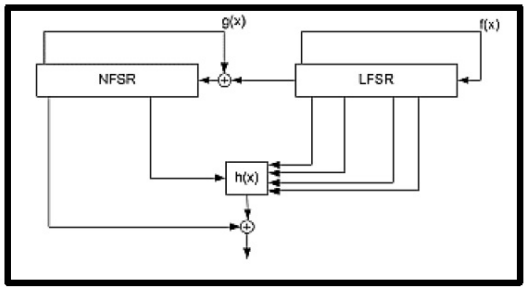

Figure 27: Grain Stream Cipher [79]

- Grain-128 proposed by M. Hell et al. in 2007 [33], a stream cipher that is very small in hardware and targets extremely limited resources environments. Its key size is 128 bits and IV size is 96 bits. Has a very simple design as shown in figure 28, based on two shift registers, one (LFSR) and one (NFSR) with an output function, the special feature of the Grain family which is considered an advantage above all other stream ciphers, is increasing the speed at the expense of adding more hardware.

  Designers of Grain-128 used the LSFR and the NFSR to guarantee good statistical properties, to ensure a lower bound for the period of the keystream and to introduce nonlinearity respectively. The nonlinear filter takes input from both shift registers LFSR and NFSR [33].

- Grain-128a proposed by M. Agren et al. in 2011, it is a new version of Grain with authentication that has built-in capability for authentication and is fortified against all known attacks and observations on the original Grain-128. the modifications on the new version are modest by keeping the basic structure of the cipher shown in figure 29, while the enhancements were made in security and this gives a high confidence in the new version Grain-128a and allows for easy

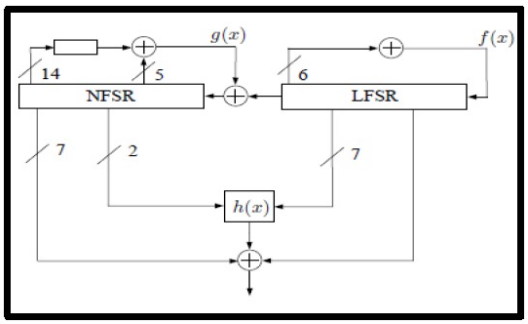

Figure 28: Grain-28 Stream Cipher [79]

updating of existing implementations. It has a key size of 128 bits and an IV size of 96 bits, well suited for hardware environments since it has a low gate count, a low power consumption and a small chip area, while designing the cipher, extra consideration was given to various attacks. The only disadvantage in Grain-128a is that is it more expensive in hardware than earlier version but surly one can ignore this for the sake of better security [2].

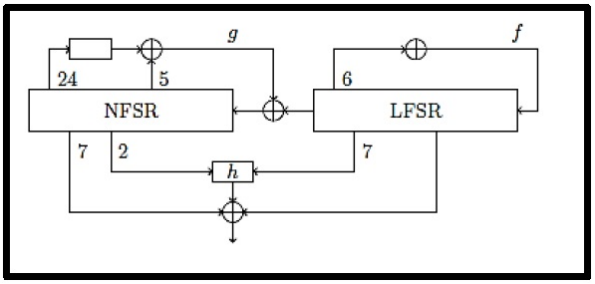

Figure 29: Grain-128a Stream Cipher [80]

- Grain-128AEAD proposed by M. Hell in 2019 [32], its quite similar to earlier version, although it has been tweaked to accommodate larger authenticators and AEAD. In addition, the modes of operation have been upgraded, all changes were made to improve the security and the protection against modern cryptanalysis, the key size is 128-bit key and IV is 96-bit and produces a pseudo random sequence that is used for messages encryption and authentication [**?**].

  Grain-128AEAD design consists of two main building blocks as shown in figure 30. The first is a pre-output generator and a pre-output function, the first is designed using a (LFSR) and a (NFSR), while the second is an authenticator generator containing of a shift register and an accumulator [**?**]. Grain-128AEAD was found competitive according to the benchmarking results of NIST, competitive performance in software and hardware have been presented in [48, 68] respectively.

- GRAIN-128AEADv2 is the last member of the Grain family, presented by M. Hell in 2019, compared to the previous versions, Grain-128AEAD regulates the cipher initialization such that the key is re-introduced at the end of the initialization in order not to allow the secret key to be immediately reconstructed in case of knowing the states of the LFSR and NFSR, [32] The Lizard stream cipher was an inspiration for this feature.

  In order to reduce key information leaking, the number of startup stages has been increased, that would result in the specifications of the updated algorithm Grain-128AEADv2. Figure 31

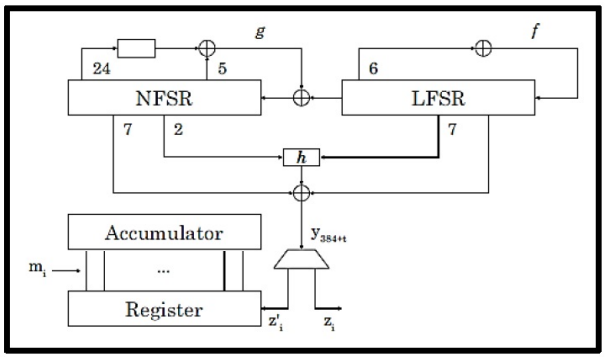

Figure 30: Grain-128AEAD Stream Cipher [41]

shows Grain-128AEADv2 Stream Cipher.

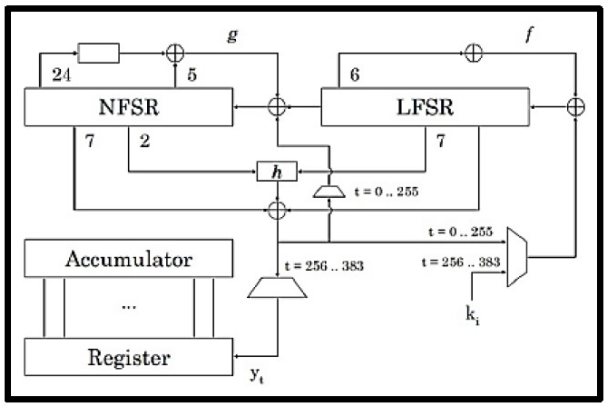

Figure 31: Grain-128AEADv2 Stream Cipher [83]

5. Comparison of the Lightweight Encryption Algorithms

Here an evaluation and comparison of the encryption algorithms implementations as they are reported in this paper. Table 1 summarize the features of the examined lightweight block cipher and Table 2 summarize the features of the examined lightweight stream Ciphers.

6. Conclusions

It is obvious that lightweight cryptography is one of the most important things in securing embedded systems, it is mostly used in constrained devices such as RFID tags, smart homes, fire detectors and other IoT applications, since using regular cryptographic algorithms for IoT devices is not ideal, therefor, Lightweight cryptographic algorithms are one of the most ideal approaches for safeguarding those IoT applications.

This paper presents the state-of-the-art implementations in lightweight symmetric-key (Block and Stream) cryptography and also many latest developments in that field, we summarize 18 block ciphers and 6 stream ciphers, 4 of them are family ciphers, all their versions have been summarized and covered from many points of view (hardware implementations, software implementations, security, cost and performance).

All the features of all the different proposed ciphers were considered and a comparative analysis was presented, with the information combined in the corresponding tables, with novel algorithms and

Table 1: lightweight Block Ciphers

Cipher	Type of Structure	Block Size (bits)	Key Size (bits)	Rounds
mCrypton [5]	SPN	64	64, 96 or 128	12
Present [15]	SPN	64	80 or 128	31
Puffin-2 [24]	SPN	64	80	34
SLIM [28]	Feistel	32	80	32
Klein [29]	SPN	64	64, 80 or 96	12, 16 or 20
SEA [31]	Feistel	N (where n= 48, 96, 144 etc.)	n (where n= 48, 96, 144 etc.)	Depends on an equation ($nr = 3n/4 + n/b + 2 \times (b/2)$)
Clefia [20]	GFN (Generalized Feistel Network)	128	128, 192 or 256	18, 22 or 26
LBlock [79]	Feistel	64	80	32
TWINE [72]	GFN (Generalized Feistel Network)	64	80 or 128	36
μ^2 [81]	GFN (Generalized Feistel Network)	64 block	80 key	15 rounds
ANU [8]	Feistel	64	128	25
ANU-II [23]	Feistel	64	80 or 128	25
NLBSIT [3]	Both Feistel and SPN	64	64	Few no. of Rounds
Piccolo [63]	GFN (Generalized Feistel Network)	64	80 or 128	25 or 31
BORON [9]	SPN	64	80 or 128	25
RECTANGLE [83]	SPN	64	80 or 128	25
LICI [57]	Feistel	64	128	31
QTL [40]	GFN (Generalized Feistel Network)	64	64 or 128	16 or 20

cryptanalysis techniques being proposed in the literature. Lightweight ciphers are high in demand nowadays, one can consider it as a trend, which is expected to grow during the coming years due to the evolution in IoT systems. Hence, hoping that this comprehensive survey will contribute in the evolution of science.

References

[1] B. Aboushosha, R.A. Ramadan, A.D. Dwivedi, A. El-Sayed and M.M. Dessouky, *SLIM: A lightweight block cipher for internet of health things*, IEEE, 8 (2020) 203747–203757.

[2] M. Ågren, M. Hell, T. Johansson and W. Meier, *A new version of Grain-128 with authentication*, Int J Wireless Mobile Comput 5(1) (2011) 48–59.

[3] A. Alahdal, G.A. AL-Rummana, G.N. Shinde, and N.K. Deshmukh, *NLBSIT a new lightweight block cipher design for securing data in IoT devices*, Int. Comput. Sci. Eng. 8(10) (2020).

[4] V. Aminghafari and H. Hu, *Fruit: Ultra-lightweight stream cipher with shorter internal state*, IACR Crypt. ePrint Arch. 2016 (2016) 355.

[5] V. Aminghafari and H. Hu, *Fruit-80: A secure ultra-lightweight stream cipher for constrained environments*, Entropy 20(3) (2018) 180.

Table 2: lightweight Stream Ciphers

Cipher	Construction	Key Size (bits)	Initialized-Vector (IV) (bits)
LOGIC [27]	2-NFSR + a chaotic module + a filter function + 3-multiplexers.	80	-
TRIVIUM [17]	Three interconnected LFSR	80	80
A4 [51]	One LFSR + One Feedback with Carry Shift Register (FCSR)	80	80
The Fruit Family (Fruit-v2 [70])	LSFR + NSFR + 2 counters	80	70
The Fruit Family (Fruit80 [74])	LFSR + NFSR + a counter	80	70
The Enocoro Family (Enocoro-80 [78])	PRNG structure	80	64
The Enocoro Family (Encoro-128v2 [55])	PRNG structure	128	64
The Grain Family (Grain ([33]))	LFSR + NFSR	80	64
The Grain Family (Grain-128 ([33]))	LFSR + NFSR + an output function	128	96
The Grain Family (Grain-128a ([2]))	LFSR + NFSR + pre-output function + a mechanism that handles the authentication.	128	96
The Grain Family (Grain-128AEAD ([?]))	Same design as the earlier version but has been tweaked to produce better authenticators and AEAD	128	96
The Grain Family (GRAIN-128AEADv2 ([32]))	Same design as Grain-128AEAD but has been improved to produce the best authenticators and AEAD	128	96

[6] J.P. Aumasson, M. Naya-Plasencia and M.-J.O. Saarinen, *Practical attack on 8 rounds of the lightweight block cipher KLEIN*, Int. Conf. Crypt. India, Springer, Berlin, Heidelberg, 2011, pp. 134–145.

[7] G.-C. Bae and K.-W. Shin, *An efficient hardware implementation of lightweight block cipher algorithm CLEFIA for IoT security applications*, J. Korea Ins. Info. Commun. Eng. 20(2) (2016) 351–358.

[8] G. Bansod, A. Patil, S. Sutar and N. Pisharoty, *ANU: An ultra lightweight cipher design for security in IoT*, Secur. Commun. Networks 9(18) (2016) 5238–5251.

[9] G. Bansod, N. Pisharoty and A. Patil, *BORON: An ultra-lightweight and low power encryption design for pervasive computing*, Front. Inf. Tech. Electr. Eng. 18(3) (2017) 317–331.

[10] D. Bellizia, G. Scotti and A. Trifiletti, *Implementation of the PRESENT-80 block cipher and analysis of its vulnerability to side channel attacks exploiting static power*, Proc. 23rd Int. Confe. Mixed Design of Integrated Circuits and Systems, MIXDES, 2016 (2016) 211–216.

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

[12] C. Blondeau and B. Gérard, *Diffrential cryptanalysis of PUFFIN and PUFFIN2*, ECRYPT Workshop on Lightweight Cryptography, (2011).

[13] C. Blondeau and K. Nyberg, *Links between truncated differential and multidimensional linear properties of block*, Adv. Crypt. EUROCRYPT 2014 LNCS, 8441 (2014) 165–182.

[14] A. Bogdanov, L.R. Knudsen, G. Leander, C. Paar, A. Poschmann, M.J.B. Robshaw, Y. Seurin and C. Vikkelsoe, *PRESENT an ultra-lightweight block cipher*, In: P. Paillier, I. Verbauwhede (eds), Cryptographic Hardware and

Embedded Systems - CHES 2007. CHES 2007, Lecture Notes in Computer Science, Springer, Berlin, Heidelberg, (2007) 450–466.
[15] M.U. Bokhari and S. Hassan, *A comparative study on lightweight cryptography*, Adv. Intell. Syst. Comput. 729 (2018) 69–79.
[16] W.J. Buchanan, S. Li and R. Asif, *Lightweight cryptography methods*, J. Cyber Secur. Technol. 1(3–4) (2018) 187–201.
[17] C.D. Cannière, *TRIVIUM: A stream cipher construction inspired by block cipher design principles*, Int. Conf. Info. Secur., 2006, pp. 171—186.
[18] M. Cazorla, S. Gourgeon, K. Marquet and M. Minier, *Survey and benchmark of lightweight block ciphers for MSP430 16-bit microcontroller*, Secur. Commun. Networks 8(18) (2015) 3564–3579.
[19] R. Chatterjee and R. Chakraborty, *A modified lightweight PRESENT cipher for IoT security*, Int. Conf. Comput. Sci., Eng. Appl. (ICCSEA), 2020, pp. 1—6.
[20] H. Cheng, H.M. Heys and C. Wang, *PUFFIN: A novel compact block cipher*, 11th EUROMICRO Conf. Digital Syst. Design Architectures, Methods and Tools, 2008, pp. 383–390.
[21] T.D. Cnudde and S. Nikova, *Securing the PRESENT block cipher against combined side-channel analysis and fault attacks*, IEEE Trans. Very Large Scale Integ. (VLSI) Syst. 25(12) (2017) 3291–3301.
[22] Y. Cui, H. Xu and W. Qi, *Improved integral attacks on 24-round Lblock and Lblock-S*, IET Info. Secur. 14(5) (2020) 505–512.
[23] V. Dahiphale, G. Bansod and J. Patil, *ANU-II: A fast and efficient lightweight encryption design for security in IoT*, Int. Conf. Big Data, IoT and Data Science, BID 2017, 2018 (2018) 130–137.
[24] A.B. Dar, M.J. Lone and N. Hussain, *Revisiting lightweight block ciphers: Review, taxonomy and future directions*, SSRN Electr. J. 2021 (2021).
[25] S. Deb and B. Bhuyan, *Performance analysis of current lightweight stream ciphers for constrained environments*, Sadhana- Acad. Proc. Engin. Sci. 45(1) (2020).
[26] S.S. Dhanda, B. Singh and P. Jindal, *Lightweight crypetography: A solution to secure IoT*, Wireless Personal Commun. 112(3) (2020) 1947–1980.
[27] L. Ding, C. Liu, Y. Zhang and Q. Ding, *A new lightweight stream cipher based on chaos—LOGIC*, Symmetry 11(7) (2019) 853.
[28] L. Ertaul and S.K. Rajegowda, *Performance analysis of CLEFIA, PICCOLO, TWINE lightweight block ciphers in IoT environment*, Proc. Int. Conf. Secur. Management (SAM), 2017, pp. 25–31.
[29] Z. Gong, S. Nikova and Y.W. Law, *KLEIN: A new family of lightweight block ciphers*, Int. Workshop Radio Frequency Identif. Secur. Privacy Issues, 2012, pp. 1—18.
[30] G. Hatzivasilis, K. Fysarakis, I. Papaefstathiou, H. Manifavas, *A review of lightweight block ciphers*, J. Crypt. Engin. 8(2) (2018) 141–184.
[31] M. Hell and T. Johansson, *Security evaluation of stream cipher Enocoro-128v2*, Access mode https://lup.lub.lu.se/luur/download, (2008).
[32] M. Hell, T. Johansson, A. Maximov, F.H.N.W. Willi Meier, S.J. Sönnerup and H. Yoshida, *Grain-128AEADv2-A lightweight AEAD stream cipher*, Information Technology, Laboratory COMPUTER SECURITY RESOURCE CENTER, (2019).
[33] M. Hell, T. Johansson and F.H.N.W. Willi Meier, *Grain: A stream cipher for constrained environments*, Int. J. Wireless Mobile Comput. 2(1) (2007) 86—93.
[34] H.K. Hoomod and A.A. Ali, *New technique for internet of things security based on the hybrid mcrypton-blowfish and chaotic system*, Int. J. Sci. Res. (IJSR) 8(8) (2019) 650–652.
[35] L. Jiao, Y. Hao and D. Feng, *Stream cipher designs: A review*, Sci. China Info. Sci. 63(3) (2020) 1—25.
[36] K.B. Jithendra and T.K. Shahana, *New biclique cryptanalysis on full-round PRESENT-80 block cipher*, SN Comput. Sci. 1(2) (2020).
[37] T. Kußmaul, J. Löffler and A. Wiesmaier, *Block ciphers PRESENT and SEA incomparision*, Technische Univ. Darmstadt, Darmstadt, Germany, Tech. Rep. TUD-CS-2016-14739, (2015).
[38] A. Kuznetsov, Y. Gorbenko, A. Andrushkevych and I. Belozersev, *Analysis of block symmetric algorithms from international standard of lightweight cryptography ISO/IEC 29192-2*, 4th Int. Scientific-Practical Conf. Problems of Infocommun. Sci. Technol. PIC S and T, IEEE, 2017, pp. 203–206.
[39] V. Lallemand and M. Naya-Plasencia, *Cryptanalysis of KLEIN*, Int. Workshop Fast Software Encryption, Springer, Berlin, Heidelberg, 8540 (2015) 451–470.
[40] L. Li, B. Liu and H. Wang, *QTL: A new ultra-lightweight block cipher*, Microproc. Microsyst. 45 (2016) 45–55.
[41] Y. Li, W. Wu, X. Yu, L. Dong, *Improved integral attacks on reduced round CLEFIA*, Int. Workshop Inf. Secur. Appl. Springer, Berlin, Heidelberg, 2011, pp. 28–39.

[42] M. Li, D. Zhao, X. Tang, S. Cheng, X. Hu and L. Bao, *Hardware implementation and optimization design of lightweight RECTANGLE algorithm*, IEEE 9th Joint Int. Inf. Technol. Artific. Intell. Conf. 9 (2020), pp. 1447–1450.
[43] H. Liang and M. Wang, *Cryptanalysis of the lightweight block cipher BORON*, Secur. Communi. Networks 2019 (2019).
[44] C.H. Lim and T. Korkishko, *MCrypton-A lightweight block cipher for security of low-cost RFID tags and sensors*, Int. Workshop Inf. Secur. Appl. Springer, Berlin, Heidelberg, 2005, pp. 243–258.
[45] I. Makarenko, S. Semushin, S. Suhai, S.A. Kazmi, A. Oracevic and R. Hussain, *A comparative analysis of cryptographic algorithms in the internet of things*, 3rd Int. Sci. Tech. Conf. "Modern Network Technologies 2020", MoNeTeC 2020 – Proc. 2020, pp. 1–8.
[46] H. Mala, M. Dakhilalian and M. Shakiba, *Cryptanalysis of mCrypton-A lightweight block cipher for security of RFID tags and sensors*, Int. J. Commun. Syst. 25(4) (2012) 415–426.
[47] C. Manifavas, G. Hatzivasilis, K. Fysarakis and Y. Papaefstathiou, *A survey of lightweight stream ciphers for embedded systems*, Secur. Commun. Networks 9(10) (2016) 1226–1246.
[48] A. Maximov and M. Hell, *Software evaluation of Grain-128AEAD for embedded platforms*, IACR Cryptol. ePrint Arch. 2020 (2020) 659.
[49] K.A. McKay, L. Bassham, M.S. Turan and N. Mouha, *NISTIR 811 national institute of standards and technology report on lightweight cryptography*, National Institute of Standards and Technology, 2017.
[50] K.S. Mohamed, *New Frontiers in Cryptography*, Springer, USA, 2020.
[51] N.A. Mohandas, A. Swathi, R. Abhijith, A. Nazar and G. Sharath, *A4: A lightweight stream cipher*, 5th Int. Conf. Commun. Electron. Syst. (2020) 573–577.
[52] 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. Network Comput. Appl. 58 (2015) 73–93.
[53] M.F. Mushtaq, S. Jamel, A.H. Disina, Z.A. Pindar, N.S.A. Shakir and M.M. Deris, *A survey on the cryptographic encryption algorithms*, Int. J. Adv. Comput. Sci. Appl. 8(11) (2017) 333–344.
[54] K. Ngo, E. Dubrova and M. Moraitis, *Bitstream modification of trivium how to attack and how to protect*, IACR Cryptol. ePrint Arch, 2020.
[55] T. Owada, K. Okamoto, Y. Igarashi and T. Kaneko, *Update on enocoro stream cipher*, Int. Symp. Inf. Theory Appl. IEEE, 2010, pp. 778–783.
[56] J.H. Park, *Security analysis of mcrypton proper to low-cost ubiquitous computing devices and applications*, Int. J. Commun. Syst. 22(8) (2009) 959–969.
[57] J. Patil, G. Bansod and K.S. Kant, *LiCi: A new ultra-lightweight block cipher*, Int. Conf. Emerg. Trends Innov. ICT (ICEI), 2017, pp. 40–45.
[58] C. Pei, Y. Xiao, W. Liang and X. Han, *Trade-off of security and performance of lightweight block ciphers in industrial wireless sensor networks*, Eurasip J. Wireless Commun. Network. 2018(1) (2018) 1–18.
[59] T.Y. Rahiyanath, *A novel architecture for lightweight block cipher*, Piccolo, Int. J. Res. Eng. Tech. 4(09) (2015) 97–103.
[60] M. Rana, Q. Mamun and R. Islam, *Current lightweight cryptography protocols in smart city IoT networks: A survey*, arXiv preprint arXiv:2010.00852, (2020).
[61] V. Rijmen, *Stream ciphers and the ESTREAM project*, The ISC Int. J. I. Secur. 2(1) (2010) 3–11.
[62] S. Sadeghi, N. Bagheri and M.A. Abdelraheem, *Microprocessors and microsystems cryptanalysis of reduced QTL block cipher*, Microproc. Microsyst. 52 (2017) 34–48.
[63] K. Shibutani, T. Isobe, H. Hiwatari, A. Mitsuda, T. Akishita and T. Shirai, *Piccolo: An ultra-lightweight blockcipher*, Int. Workshop Crypt. Hardware Embedded Syst. 2011, pp. 342–357.
[64] T. Shirai, K. Shibutani, T. Akishita, S. Moriai and T. Iwata, *The 128-bit blockcipher CLEFIA (extended abstract)*, FSE 2007, LNCS, 4593 (2007) 181–195.
[65] N. Shrivastava, P. Singh and B. Acharya, *Efficient hardware implementations of QTL cipher for RFID applications*, Int. J. High Perform. Syst. Arch. 9(1) (2020) 1–10.
[66] 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. Humanized Comput. 2017 (2017) 1–18.
[67] L. Sliman, T. Omrani, Z. Tari, A.E. Samhat and R. Rhouma, *Towards an unltra-lightweight block ciphers for IoT*, J. Info. Secur. Appl. 61 (2021).
[68] J. Sönnerup, M. Hell, M. Sönnerup and R. Khattar, *Efficient hardware implementations of Grain-128AEAD*, Int. Conf. Crypt. India, Springer, 2019, pp. 495–513.
[69] Sony Corporation, *The 128-bit blockcipher CLEFIA algorithm specication*, ReVision, 1 (2007) 1–41.
[70] A. Srivastava and A. Kumar, *A review on authentication protocol and ECC on IoT*, Int. Conf. Adv. Comput.

Innov. Technol. Engin. (ICACITE), 2021, pp. 312–319.
[71] F.-X. Standaert, G. Piret, N. Gershenfeld and J.-J. Quisquater, *SEA: A salable encryption algorithm for small embedded applications*, Int. Conf. Smart Card Res. Adv. Appl., Springer, 2006, pp. 222–236.
[72] T. Suzaki, K. Minematsu, S. Morioka and E. Kobayashi, *Twine: A lightweight, versatile block cipher*, ECRYPT workshop on lightweight Cryptography, 2011 (2011).
[73] R. Syal, *A comparative analysis of lightweight cryptographic protocols for smart home*, Adv. Intell. Syst. Comput., 882 (2019) 663–669.
[74] C. Tezcan, *The improbable differential attack: Cryptanalysis of reduced round CLEFIA*, Int. Conf. Crypt. India, Springer, Berlin, Heidelberg, 2010, pp. 197–209.
[75] V.A. Thakor, M.A. Razzaque and M.R.A. Khandaker, *Lightweight cryptography algorithms for resource-constrained IoT devices a review comparison and research opportunities*, IEEE Access 9 (2021) 28177–28193.
[76] Y. Tian, G. Chen and J. Li, *On the design of trivium*, Beijing Daxue Xuebao (Ziran Kexue Ban)/Acta Scientiarum Naturalium Universitatis Pekinensis 46(5) (2010) 691–698.
[77] C. Wang and H.M. Heys, *An ultra compact block cipher for serialized architecture implementations*, Canad. Con. Electr. Comput. Eng. (2009) 1085–1090.
[78] D. Watanabe, K. Ideguchi, J. Kitahara, K. Muto, H. Furuichi and T. Kaneko, *Enocoro-80: A hardware oriented stream cipher*, Third Int. Conf. Availab. Reliab. Secur. IEEE, 2008, pp. 1294–1300.
[79] W. Wu, L. Zhang, *LBlock: A lightweight block cipher*, Int. Conf. Appl. Crypt. Network Secur. Springer, Berlin, Heidelberg, 2011, pp. 327–344.
[80] X. Xia, B. Chen and W. Zhong, *Correlation power analysis of lightweight block cipher algorithm LiCi*, J. Phys. Conf. Ser. 1972(1) (2021).
[81] W.Z. Yeoh, J.S. Teh and M.I.S.B.M. Sazali, *µ2: A lightweight block cipher algorithm LiCi*, J. Phys.: Conf. Ser. 1972(1) (2021).
[82] M. Yoshikawa, Y. Nozaki and K. Asahi, *Electromagnetic analysis attack for a lightweight block cipher TWINE*, 2016 IEEE/ACES Int. Conf. Wireless Inf. Technol. Syst. Appl. Comput. Electrom.(ACES), IEEE, 2016, pp. 1–2.
[83] 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 Info. Sci. 58(12) (2015) 1–15.