IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 23, Issue 2, Ser.III (Mar. – Apr. 2021), PP 01-14
www.iosrjournals.org

Lightweight Cryptographic Techniques in Internet of Things: A Review

AsjahUlAin
Department of Computer Sciences Sant Baba Bagh Singh University

Er. HarpreetKour
Department of Computer Sciences Sant Baba Bagh Singh University
Jalandhar - 144030

Abstract:
Lightweight cryptography, the modern branch of cryptography is used in wireless sensor network devices or Internet of Things (IoT) devices. Lightweight cryptography came into existence with the development in IoT devices and was basically used to overcome the constrained environment conditions, e.g. RFID tags, sensors, healthcare devices. The motivation of lightweight cryptography is to use less memory, fewer computing resources, and less power supply with minimal implementation cost. The main aim of the research will be to go through an exhaustive study of lightweight cryptography.

Keywords*: Lightweight cryptography, security, privacy, confidentiality*

Date of Submission: 29-03-2021 Date of Acceptance: 12-04-2021

I. Introduction:

The term "Internet of things" also abbreviated as IoT was coined by Kevin Ashton in late 1998. It is currently the trending topic in the field of research. The Internet of Things is a new trend that is revolutionizing computing. The Internet of Things (IoT) is a recent trend that extends the boundary of the Internet to include a wide variety of computing devices. It is tailored intending to connect all objects around us with the internet, providing "anytime, anywhere" access to data. For example, residents of a house can easily control the devices with wireless connectivity at home [1]. Some more examples include wearables like fitness trackers and smart watches, smart electric power grid, smart cars, smart cities, smart medical devices, and many more. It is expected that by the end of 2020 around 50billion things will be connected to the internet. With IoT (internet of things) we can connect our household things to the internet and access them remotely without getting in physical contact with them. As more devices become connected to the Internet, there are more and more concerns about access data and also the security of IoT. As a huge amount of data is stored in the air each day, it becomes vulnerable to different types of attacks like man-in-the-middle, cloud jacking, differential attack, vehicle cyber-attack, denial of services attack, etc. With the emergence of such technology, security and privacy have become the primary concern [2]. Various solutions can be used at different layers to provide end-to-end security. Cryptography is employed to secure data. It is actually a technique used for encryption of data and hence transmits it securely. Cryptographic techniques are of two types: symmetric and asymmetric. In symmetric key encryption, a single key is used for encryption and decryption, while in asymmetric key encryption two different keys, private and public keys are used for communication between two different parties. A public key is used by the sender for encryption of data, while the receiver uses its private key for decryption of this data [3]. In terms of the security of the Internet, the security framework of the Internet cannot provide a complete solution to solve all security problems in IoT. As IoT devices have many constraints and limitations in terms of power resources, computational resources, and even memory, conventional cryptographic techniques and algorithms that we apply on high-end devices cannot apply to IoT devices and hence security measures like encryption, authentication, access control, network security and application security for IoT devices remains ineffective. Therefore, existing cryptographic methods should be enhanced by the overall IoT ecosystem effectively.

Lightweight cryptography, the modern branch of cryptography, is used in wireless sensor network devices or IoT devices. The motivation of lightweight cryptography is to use less memory, fewer computing resources, and less power supply with minimal implementation cost. It also minimized the overall implementation cost of cryptographic primitives without compromising security, it basically focuses to optimize the encryption algorithms that are based on conventional cryptographic techniques to run in are resource constraint environments.

This paper discusses the security mechanism in IoT devices, particularly focusing on lightweight cryptography. Conventional cryptography is discussed followed by its demerits in low-power IoT applications. The importance of Lightweight cryptography, a modern branch of cryptography, is described in detail. An exhaustive survey of lightweight cryptographic procedures and their implementation is also described comprehensively. The lightweight cryptographic primitives including block ciphers, stream ciphers, and hash functions available in the market are thoroughly presented and illustrated along with their implementation.

Section 2 outlines the related work to the survey. It includes some survey papers and also explains why our research is better than the previous researches. Section 3 describes lightweight cryptography with subsections explaining its types. Figure 1 illustrates it diagrammatically. It also explains various cryptographic primitives

II. Related Work:

Bhardwajet al. in 2017 made a comparative analysis that showed how lightweight cryptographic algorithms show excellent performance in terms of power consumption, memory requirements when compared to conventional cryptographic techniques. This paper also discusses different architectures of IoT, their security and privacy issues, and how lightweight cryptography can be used to resolve them [4]. SushaSurendranet al. in year 2018 discussed some lightweight cryptographic algorithms in their paper 'A Survey of Cryptographic Algorithms for IoT Devices'. Also, some possible attacks on these ciphers are also studied and performances of some ciphers on windows and embedded platforms are performed [5]. Indira KalyanDuttaet al. in year 2019 surveys solutions of lightweight cryptography in 'Lightweight Cryptography for Internet of Insecure Things: A Survey'. It covers some solutions and also gives a comparison between different types of block ciphers with recent approaches of AES [6]. Abdul Razzaqet al. in March 2020 presented a thorough survey of lightweight cryptography in paper ' A systematic technical survey of lightweight cryptography on IoT environment'. Comparison and performance analysis of some block ciphers and hash functions is also discussed along with some important features like throughput, latency, power, energy etc [7]. Suzan Sallamet al. in year 2018 made a survey on lightweight cryptographic primitives in their paper 'A survey on lightweight cryptographic algorithms'. In this paper comparison between various selected lightweight algorithms is carried out. Block ciphers and stream ciphers, their security and hardware implementations is also discussed [8]. Sumit Singh Dhanda in the year 2020 made a survey of lightweight cryptographic algorithms that are available up-to year 2019. 54 lightweight cryptographic primitives along with their comparisons is discussed. Few open research problems of LWC are also highlighted [9]. Ankit Shah et al. in year 2018 presented a survey paper 'A Survey of Lightweight Cryptographic Algorithms for IoT-Based Applications'. A comprehensive study is done on various algorithms, and data provided in this paper can be used to decide which particular algorithm can be used for a particular application. This paper also states that majority of the lightweight cryptographic primitives depend on hardware so they perform normally even with ultra-constrained devices [10]. Saurabh Singh et al.presented a paper 'Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions' in year 2017 in which existing Lightweight cryptographic algorithms are discussed. Also cryptographic primitives based on key size, block size, structure and number of rounds are discussed in detail [11]. Merly Annie Philip et al. in 2017 discussed some existing block ciphers like PRESENT, KATAN, Hummingbird, SIMON and RECTANGLE and stream ciphers TRIVIUM, GRAIN, CHACHA, WG-8 and ESPRESSO in paper 'A Survey On Lightweight Ciphers For IoT Devices' [12]. George Hatzivasiliset al. in 2018 in their paper 'A review of lightweight block ciphers' carried a survey on existing lightweight cryptographic primitives and specifically examined lightweight implementations of symmetric key algorithms in HW and SW architectures and also highlighted 52 block ciphers and 360 implementations [13]. Sattar B. Sadkhan in year 2018 in paper 'A Survey on Lightweight-Cryptography Status and Future Challenges' discussed several algorithms that are integrated for complete security of a system. During implementation various features are analyzed [14].

This work gives an exhaustive idea of lightweight cryptography in context to low power IoT and wireless sensor networks. Lightweight cryptographic primitives including block ciphers, stream ciphers and hash functions are discussed in detail. The challenges and perceptions regarding lightweight primitive functions are also deliberated in the survey. This survey also provides research ideas, domains, and challenges faced by the research community in lightweight cryptography. The survey covers almost all modern lightweight cryptographic techniques with an extensive comparison on the basis of structure and performance metrics.

III. Lightweight Cryptography

Lightweight cryptography is used to optimize encryption algorithms to run on devices with a constrained environment. As the world is becoming a global village, more and more devices are being connected to the internet day by day. With the increasing number of these devices, security is becoming the primary concern, so cryptographic primitives came into existence. As conventional cryptographic primitives need more storage and processing resources, hence these algorithms become inefficient for IoT devices. So the

advanced version of cryptographic techniques called as lightweight cryptography was modeled in 1999.Lightweight cryptography was basically made for resource constrained devices like RFID tags and WSN where conventional cryptography couldn't work.

3.1 Lightweight cryptographic primitives

In this chapter, some lightweight cryptographic primitives are discussed and summarized based on their different parameters.

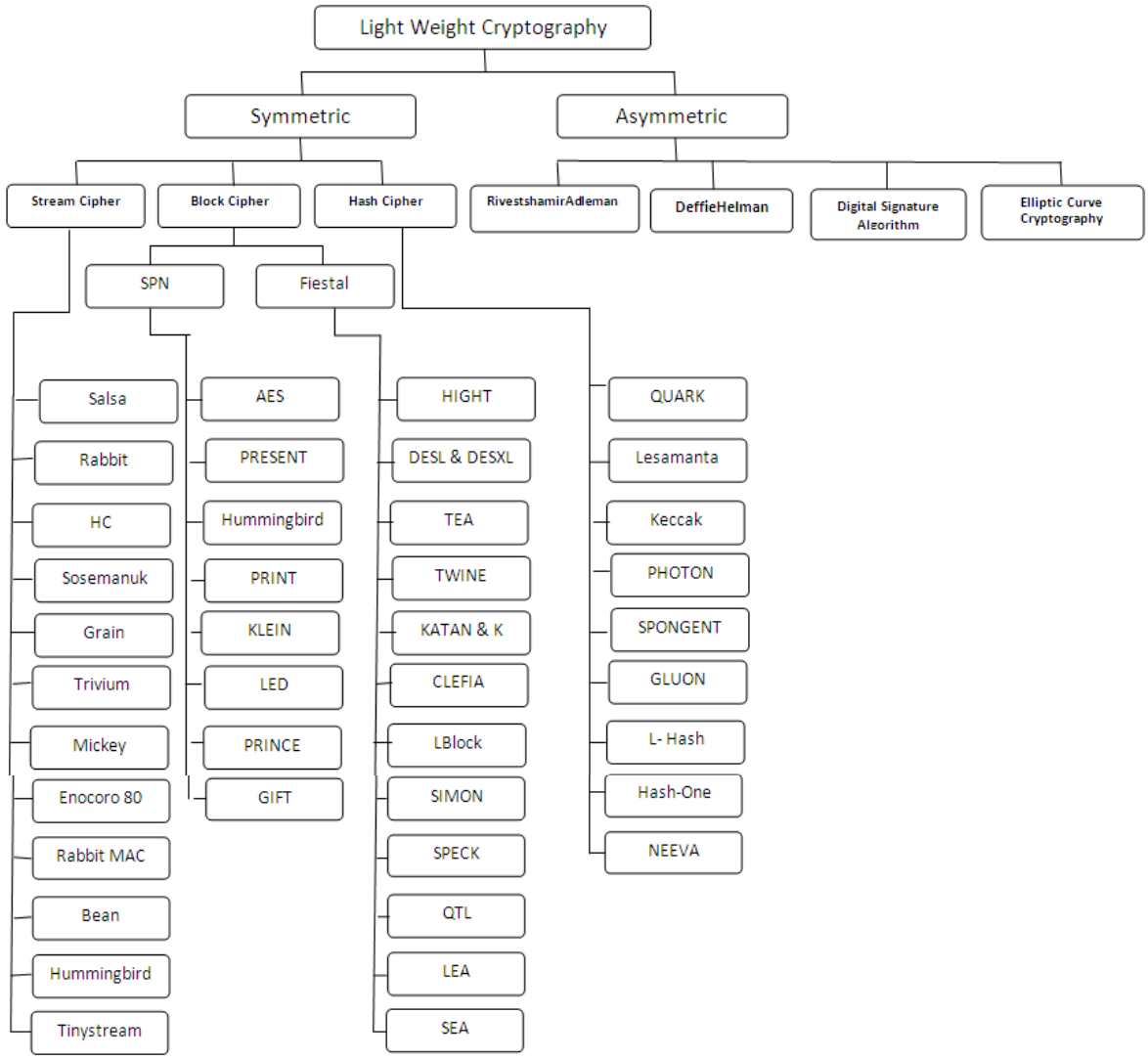


Figure 1

It is divided into two types: symmetric and asymmetric. The symmetric algorithm also called Secret key encryption uses a single key for encryption and decryption of data while the asymmetric algorithm that is also called as public-key algorithm uses two different keys for encryption and decryption where the public key is used by send and the private key is used by receiver [15]. Symmetric cryptography is further divided into a stream cipher, block cipher, and hash functions while asymmetric cryptography is divided into Rivest-Shamir Adleman (RSA), Diffie-Hellman, Digital Signature Algorithm (DSA), Elliptical Curve Cryptography (ECC). Asymmetric cryptography also provides authentication as well as non-repudiation.

A stream cipher is a symmetric key algorithm where the key size used is the same as that of data. Here "bit by bit" operation is carried out on plain text and hence ciphertext is obtained. This method is rarely used in modern cryptography. A5/1, Grain, Trivium, Rabbit are some first-generation lightweight stream ciphers. A5/1 was given in 1987 and was used for GSM [16] and Rabbit [17] in 2003. Estream, a project that was set up in November 2004 and continued till 2008 after the failure of six stream ciphers that were presented to NESSIE (New European Schemes for Signatures, Integrity, and Encryption) project presented a portfolio of stream ciphers in April 2008. This portfolio of ciphers was divided into two categories, Profile 1 contain stream ciphers that are more suitable for software applications and with high throughput requirements, and Profile 2 with

stream ciphers that are suitable for hardware applications with restricted resources like limited resources, gate count, power consumption [18]. After checking them for cryptanalysis and security vulnerabilities, several ciphers were accepted and some of them were rejected. The finalist in Profile 1 portfolio includes Salsa20/12 [19], Rabbit [20], HC-128 [21] and SOSEMANUK [22], while Profile 2 portfolio included Grain [23], Trivium [24] and MICKEY 2.0 [25].

Profile 1:

1. Salsa20/r: where 'r' denotes the number of iterations for round function, this stream cipher was proposed by D.J. Berenstein in 2005. This stream cipher used 128 and 256-bit key size, and an initialization vector with 128 bit. 12126 GE chip area is required for its hardware implementation. Its structure is based on Add-Rotate-XOR and works on the 32-bit word.
2. Rabbit: It was presented in 2003. Rabbit occupied 3800 GE and achieved 88Kbps of throughput [26]. It uses a 128-bit key size and 64-bit initialization vector.
3. HC: It has two variants HC-128 uses a 128-bit key and HC-256 that uses a 256-bit key with a 128-bit initialization vector [27]. This cipher is best suited for parallel computing, where execution is carried out simultaneously. It works well in the software when a large stream of data is to be encrypted as it is a table-driven cipher and uses two large secret tables for a 32-bit word. However, the performance decreases with the small data stream. To cover memory requirements in hardware, 52400 GE would be occupied [28]. This cipher is considered as safe as no significant cryptanalysis has been reported on it [29,30].
4. SOSEMANUK: Its structure is based on Linear Feedback Shift Register (LFSR) and Finite State Machine (FSM) with key size lying between 128 and 256 bit and initialization vector size 64 and 128 bit depending on the key size. Its implementation occupies GE of 2700 for 128 and 4100 for a 256-bit key. It uses some features of stream cipher SNOW 2.0[31] and blocks cipher Serpent [32]. Its implementation in software takes about 2000-5000 bytes of code and can achieve a throughput of 360Kbps [33], while in hardware its implementation occupies 18819GE and achieves a throughput of 3200Kbps [34]. Around 223.6-byte storage, 235.16 SOSEMANUK iterations, and 4608 faults are required by differential fault analysis attack to recover the inner state [35]. A Pc requires 11.36 hours to perform this attack.

Profile 2:

1. GRAIN: Its structure uses both Linear Feedback Shift Register (LFSR) as well as non-linear filtering function. This cipher is used for lower memory, limited gate count, and low power consumption environment. It uses an 80-bit key size and 64-bit initialization vector. For 1bit/cycle it required about 1294 GE and for 16bit/cycle it requires 3239 GE [36]. Grain 128 is a variant of Grain cipher that uses a 128-bit key and 96-bit initialization vector. It was basically designed for the applications that required high security. It uses word length from 1bit to 32 bits. For 1bit/cycle it uses 1857 GE and 32 bit uses 4617 GE. Here a smaller number of faults are injected by the attacker and the secret key is recovered.
2. Trivium: It is an asynchronous, bit-oriented stream cipher that uses a key size of 80 bit and initialization vectors of 80 bit [37]. It was basically designed to check how far can be stream ciphers modified without compromising its security or any other features like speed and flexibility. Its implementation requires 2017 GE in hardware in standard Complementary Metal Oxide Semiconductor technology [38]. Although this stream cipher was designed for hardware implementation, it works well on software implementations also [39]. By injecting only two faults and using 420 key stream bit the inner state of the cipher can be recovered by an improved differential fault analysis attack [40].
3. MICKEY: MICKEY also known as Mutual Irregular Clocking KEY stream generator uses 80-bit key size and initialization vector value can vary from 0 to 80bit and it can generate 240 key stream bits from each pair of key and IV. In a hardware implementation, 3188GE is occupied [41]. Like Grain and Trivium, a differential fault attack was demonstrated in [42]. As MICKEY 2.0 is more complex than Grain and Trivium, so attack requires 214.7 faults.

MICKEY has another variant known as MICKEY-128 2.0[43] that was designed to provide higher security. It uses a key size of 128 bits, and IV keeps varying from 0 to 128 bits. Its performance is suitable for hardware and 5039 GE is required for its hardware implementation. Like MICKEY 2.0 related key attacks are the likely attacks on MICKEY-128 2.0[44].

Other stream ciphers include Enocoro 80[45], Rabbit-MAC [46], BEAN [47], Hummingbird [48], Tinystream [49], Hummingbird-2[50], Quavium [51], Cavium [52], CAR30[53], ALE [54], ACRON [55], Sprout [56], Fruit [57], Plantlet [58], Espresso [59], lizard [60].

Cryptographic Primitive	***Key Size***	***Area (GE)***	***IV***	***Type***	***Internal State***	***Throughput (Mbps)***	***CMOS Process***
Salsa 20/r [19]	128,256	12126	128	ARX	512 bits	990	130nm
Rabbit [20] [26]	128	3800,4100	64	Chaotic table, simple athematic	513 bits	0.080	180nm
HC [21] [27] [28] [29] [30]	128		128				
SOSEMANUK [22] [31] [32] [33] [34] [35]	128,256	4100 2700,18,819	64,128	LFSR & FSM	-	800	0.09nm/90nm
GRAIN [23] [36]	80,128	1857 4617	96	LFSR&NLFSR		327.9,22299.6	130nm
Tirvium [24] [37]	80	2580 4921	80	3SHR			130nm
Mickey [25] [41] [42] [43] [44]	80,128	3188,5039	0-80, 0-128	Galois LFSR+NLFSR		454.5	130nm
Enocoro (128,80) [45]	128,80	4100,2700	64,64	ARX/PRNG		800	90nm
Rabbit Mac [46]							
Bean [47]	80			2-FCSR			
Humming Bird [48]	256				80 bits		
Quavium [51]	80	3496,2372	80	4-Trivium like SHR	288 bits		
Car30[53]	128		120				
ALE [54]	128	2581					
Acorn [55]	128		128				
Sprout [56]		813,839		NLFSR&LFSR+ Counter Reg		0.100	
Fruit [57]	64/80	990	64	LFSR& NLFSR		0.100	
Plantlet [58]	80	928	90	LFSR & NLFSR			180nm
Espresso [59]	128	1500	96	Galois Structure			90nm
Lizard [60]	120	1161	64 3578	NLFSR	121		180nm

Table 1: Comparison of lightweight stream cipher

The hash function is another type of symmetric-key algorithm where a random size of data is taken as input and then converted into a fixed size of a unique string of data of shorter length [61]. The string that is produced is called 'hash value' or 'message digest' or 'digital fingerprint' or 'digest' or 'checksum' [8, 61]. To achieve high efficiency and required security, a dedicated hash function usually needs more memory size, which increases the cost. So there is a trade-off between security and efficiency. The construction model Merckle-Damgard is the most popular way of constructing hash functions, especially SHA-1 and SHA-2[62].

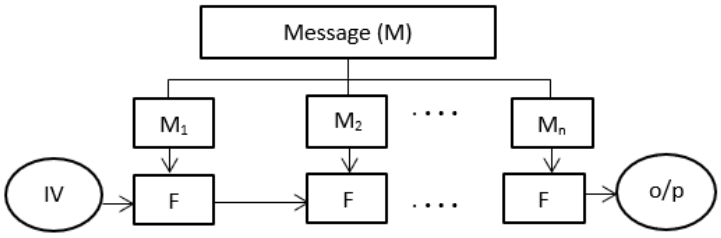


Figure 2: Merckle Damgard Construction model

Some hash functions are discussed below

1. Quark:Hash family Quark was presented by Jean-Philippe Aumasson et al. [63] in the year 2010, which was inspired by Grain (stream cipher) and KATAN (block cipher). It uses the sponge function and is defined in three instances: U-Quark, d-Quark, s-Quark. U-Quark provides 64-bit security with chip area 1379 GE while s-Quark provides 112-bit security with chip area 2296 GE and power consumption of 2.44 μWatts and 4.35 μWatts respectively.

2. Lesamanta-LW: Lesamanta, a 256-bit hash function was proposed in 2010[64]. In a hardware implementation, only 8.24KGates are required on 90nm technology. S-box is the same as in AES. In a software implementation, 50 bytes of RAM are required for an 8-Bit processor.
3. Keccak: Keccak was presented in 2010 by Kavun and Yalcin [65] and belongs to the family of cryptographic sponge functions. In 2015 Keccan became the Federal Information Processing Standards (FIPS) 202 (SHA-3) but its chip area for implementation exceeds a bit than lightweight standards. The maximum chip area required for parallel Keccak implementation is 4763 GE and for serial Keccak it requires 2079 GE. While the minimum chip area required for parallel Keccak is 409 GE and for serial it is 252 GE.
4. PHOTON: It was proposed in 2011[66]. Sponge functions are used for domain extension algorithms and it has AES like features. Also, a new mixing layer was introduced which helped to achieve collusion-resistant security of 64 bit with an area requirement of 1120 GE. It is defined as PHOTON-n/r/r' where n is hash size, r is the input bit rate, r' is the output bit rate.
5. SPONGENT: It was proposed in 2011 by [67]. It is also based on sponge construction and has got PRESENT-type permutation.
6. GLUON: This lightweight hash function was proposed byBerger et al. [69]. The GLUON family has been inspired by two stream cipher families,viz F-FCSR-V3[70] and X-FCSR-V2[71]. It is a Sponge construction based hash family. A message digest of 160 bit is produced with 80-bit security and implementation occupied is 2799 GE.
7. L-Hash: This is an ultra-lightweight hash function that was proposed in 2013[72]. L-Hash can support a message digest of 80 bits, 96 bits, and 128 bits, and in its internal permutation, it uses a Feistel-PG structure and can provide 60-bit security. With serialized implementation, it requires about 817/1028 GE.
8. Hash-One: This hash function was presented in 2016[73]. A message digest of 160 bits is produced, and it consumes a gate equivalent of 1006 with 80-bit security in terms of sponge capacity.
9. NEEVA: This sponge based is the latest lightweight hash function used in RFID technology [74].

Cryptographic Primitive	***Size (bits)***	***Area (GE)***	***Security***
Quark [63]	128,224	1379/2296	64/112 bit security
Lesamanta [64]	128	8240	120-128 bit security
Keccak [65]	160	4763-252	-
PHOTON [66]	128,256	1120	64 bit equivalent security
SPONGENT [67]	88-256	738-1950	40-128 bit security
GLUON [69] [70] [71]	160	2799	80 bit security
L-Hash [72]	96	817/1028	60 bit security
Hash-One [73]	160	1106	80 bit security/ 160 bit preimage resistance/80 bit collusion resistance
NEEVA [74]	224	-	112 bit security

Table 2: Comparison of Lightweight Hash Functions

Block cipher is the type of symmetric key algorithm in which a block of text is encrypted at a time. In this research, the focus is on symmetric block ciphers. Block ciphers are divided into two structures: SP-network structure and Feistel structure.

1. SP network: Substitution Permutation Network (SPN). Initially, the plaintext is XORed with a round key. Then four r-1 rounds, substitution operation (SubBytes) is performed on the state with each round followed by permutation ShiftRows and then again XORed with a round key.The below figure shows the SP network of three rounds.

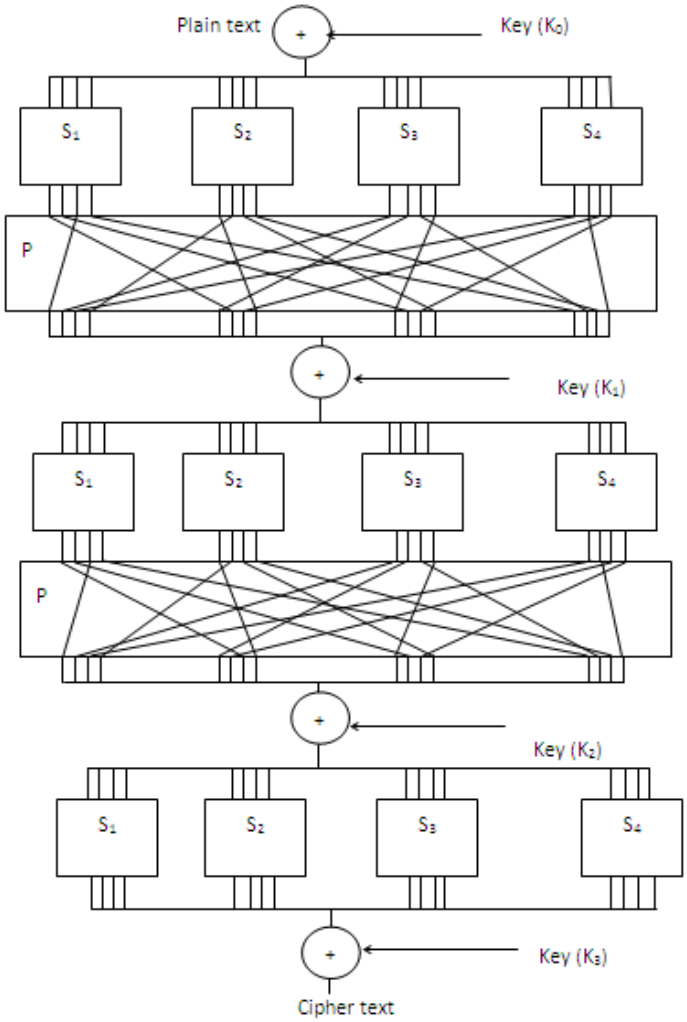


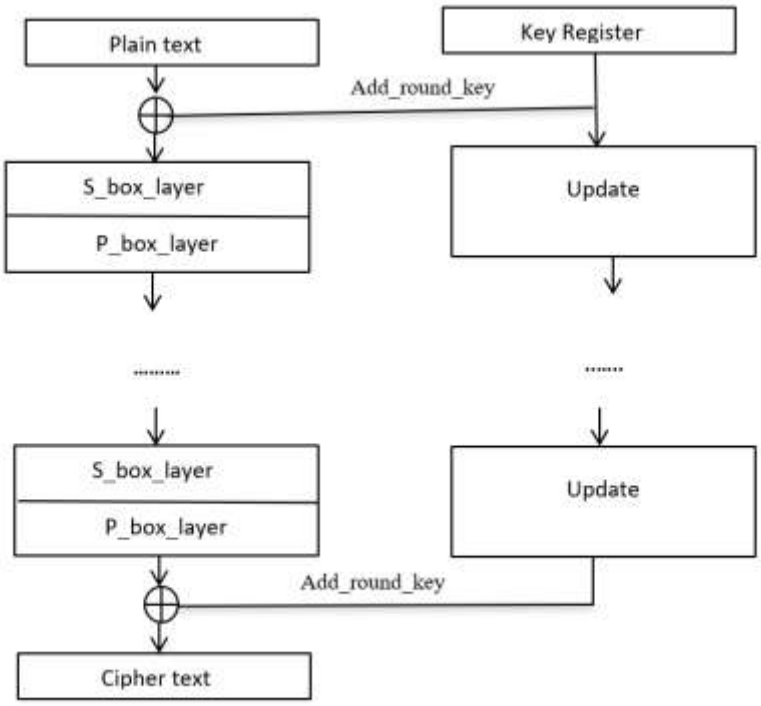
Figure 3: SP Network

1. AES: AES (Advanced Encryption Standard) is a standardized method of encrypting text that was proposed in 1998 by Joan Daemen and Vincent Rijmen [75]. It follows SPN behavior. It uses 128, 192, 256 keysize with 128-bit block size and 10, 12, and 14 rounds respectively. Except for some physical implementation attacks, bi-clique cryptanalysis AES is secure against various attacks but AES has the problem of using a huge amount of resources and processing power which makes it inefficient for some devices like RFID tags or sensor nodes that are resource-constrained devices.
2. PRESENT: This block cipher was presented by Bogdanov *et al.* [76] in year 2007. The block length is 64 bits with key size 80 or 128 bits and 32 rounds. S-box is of 4 bits and in one round it is used 16times.

XOR operation is carried out in 31 rounds to form a round key, and the last round is used for post-whitening. Each round in PRESENT is carried out in 3 steps:

a) Add_round_key: Here key is XORed with cipher (input function)
b) Substitution: It uses 4 bits S-box in parallel
c) Permutation: It uses P-layer. It is carried out to provide diffusion.

PRESENT has an SPN structure. It is used in applications where AES is unsuitable. A possible attack on PRESENT is related to key rectangle attack. The main module of the cipher is given below.



3. Hummingbird: Designed by Engels, Schweitzer, and Smith Hummingbird is a hybrid of stream cipher and block cipher that uses SPN structure with 16bit block size and 256-bit key size. When compared with block cipher PRESENT in terms of throughput for size and speed optimized implementations, Hummingbird is 147 and 4.7 times faster. It is resistant to birthday attack, differential and linear cryptanalysis [77] but can get affected by cube attack if the degree of internal state transition function is low in a stream cipher [78]. In [77] the author implements two lightweight cryptographic primitives viz PRESENT and HUMMINGBIRD on the 8-Bit and 16-bit micro-controller for size optimization and found that Hummingbird is 13% and 69% less than PRESENT.
4. PRINT: it was designed for IC printing purposes. PRINT has an SP Network that uses a message block of 48 or 96 bit, with 80 or 160-bit key size and 48 or 96 rounds respectively [79].
5. KLEIN: Gong *et al.* [80]presented this cipher. It has an SPN (Substitution-Permutation) type network behavior. The substitution layer (SubNibble) has 16 4bits S-boxes, while the permutation layer has RotateNibbles and MixNibbles. Block size is 64bits, keysize is 64, 80, or 96 bits with 12, 16, or 20 rounds respectively. Key-recovery integral attack is a possible attack on KLEIN.
6. LED: Guo *et al.* proposed block cipher known as Light Encryption Devicein 2011[81]. It has 4 variants of 64-bit keysize which use 8 rounds, 80, 96, and 128-bit keysize that use 12 rounds with block size 64. It has an SPN structure network. The meet-in-the-middle attack cannot affect this cipher, but it suffers related-key and single-key attacks. In [81] two software implementations of LED were made: One reference and clarity and the second for performance by using lookup tables, an Intel (R) Core (TM) i7 CPU Q 720 clocked at 1.60GHz were used for measurements. LED state is represented as a 64-bit word with 8 lookup tables.
7. PRINCE: PRINCE was designed to improve the overall hardware implementation. It is a 64-bit block cipher that has SPN structure and was derived from AES and PRESENT in 2012 The Technical University of Denmark (DTU), NXP Semiconductors, and the Ruhr University Bochum cooperated and proposed this block cipher algorithm. It uses a 64-bit message block and 128-bit key size with 12 rounds [82, 83].
8. GIFT: It was discovered in the year 2017 and used an SPN technique. It uses a 128-bit key size with a block size of 64 or 128 and 28 or 40 rounds. It is smaller and faster cryptographic primitive and has overcome some know the weakness of cipher PRESENT [84]. In [84], Banik S. et al. produced bit-slice implementation for AVX2 registers as this cipher has inherent bit-slice structure, and results were benchmarked on a computer with Intel Haswell processor (i5-4460U). Results of the other two ciphers, SIMON and SKINNY were also produced on the same computer.
9. Feistel Structure: A Feistel structure in symmetric cryptography is used to make block ciphers. Here the plain text undergoes multiple rounds of processing and the size of ciphertext is equal to the size of plain text.

Each round comprises a substitution step, which is followed by a permutation step. Initially, the input block is divided into two halves (L0, R0) for the left and right half. On the right half (R0), a function is applied and in the function, a sub-key is used that is generated from a master key. The output hence obtained is then XORed with left half (L0) and then their output is swapped. This completes one round of algorithm. Similarly, many rounds are carried out on the plaintext. The number of rounds carried depends on the size of plain text.

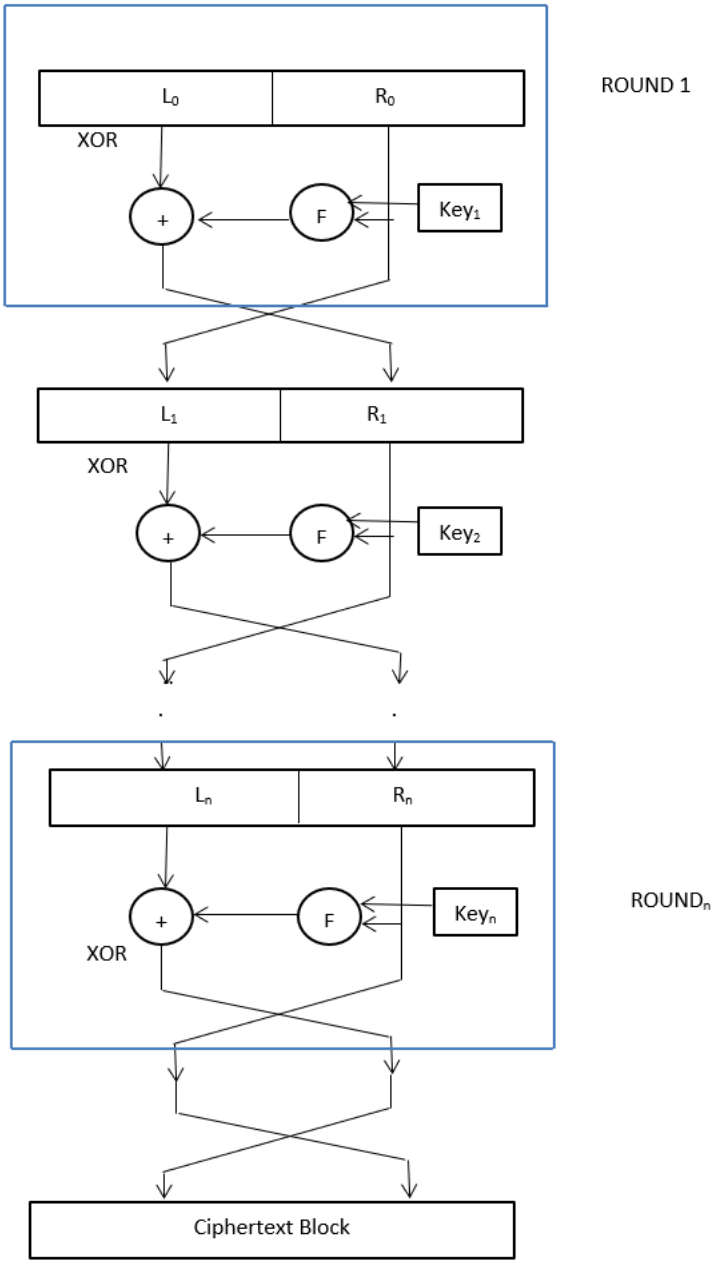


Figure: Fiestel structure

1. HIGHT: This block cipher was proposed by Hong *et al.* in 2006. It has a 64bit block length and 128bit key length. It has a Feistel-like structure with a 32 round iterative structure. HIGHT is an ultra-lightweight stream cipher that can be used for less cost, low power, and devices with fewer resources. Related key rectangle, impossible and differential attacks are the potential attacks on this cipher [85]. To check the hardware complexity on 0.25µm CMOS technology, in [85] a HIGHT circuit is designed with three parts: round function, key schedule, and control unit. 3048 NAND gates are required and throughput is about 150.6Mbps at an 80MHz clock rate.
2. DESL & DESXL: DESL is a Lightweight derivative of conventional cryptographic primitive Data Encryption Standard (DES). It is a symmetric block cipher that uses a 64-bit block size with 16 rounds. Each round is a Feistel round.

DESXL is a lightweight version of conventional cryptographic primitive DESX [86].
3. TEA: David Wheeler and Roger Needham introduced TEA lightweight block cipher in 1994. It uses a 128bit key and operates on a block size of 64bit. TEA has aFeistel structure with 64 rounds. Although TEA is simple to implement, it suffers from an equivalent key problem i.e. each key is equivalent to the other three keys making it prone to attacks. This weakness of TEA led to the design of the XTEA cipher. Like TEA which operates on 32-bit unsigned integers, XTEA also operates a 64bit block with a 128bit size key, it has also got a Feistel structure and also with 64 rounds. But XTEA also suffers from two attacks,i.e. related key differential impossible differential cryptanalysis attacks [87].
4. TWINE: This cipher was proposed in the year 2011. It is a 64-bit block cipher with 80 and 128bit key lengths. TWINE-80 or TWINE-128 is used to denote key length with 32 rounds. It has a Feistel structure. Biclique Cryptanalysis is the possible attack on this cipher. Biclique Cryptanalysis is the Meet-In-The-Middle recently proposed variant [88].
5. KATAN & KTANTAN: KATAN was proposed in 2009. This cipher has three variants: KATAN-32, KATAN-48, and KATAN- 64 with 80-bit keysize and 254 rounds. KATAN & KTANTAN families can withstand linear attacks and differential attacks but are not resistant against slide attacks. This attack is a possible attack on KATAN & KTANTAN. KTANTAN families are prone to Cube Attacks and Algebraic Attacks [89].
6. CLEFIA: It was developed by Sony, proposed by Shirai et al. [90]. CLEFIA has got a generalized Feistel structure that has a block size of 128 bits and uses a key size of 128 bits or 192 bits or 256 bits.
7. Lblock: LBlock was proposed in 2011at ACNS by Wu and Zhang. It uses a 64bit block size, 80bit key, and 32 rounds [91]. M. N. Hasanet al. in [92] implemented the LBlock lightweight block cipher in the Altera DE1 FPGA board, the encryption required 32 clock cycles to encrypt data of 64-bits. It was observed that 326 Logic Elements (LEs) in the Cyclone II EP2C20F484C7 platform are required. It works on the maximum clock frequency of 156.32 MHz with thermal power dissipation of 124.06 mW, which is better than XTEA, Hummingbird, and KATAN_64.
8. SIMON: It was proposed by the National Security Agency (NSA) in June 2013. It was used to improve the performance in hardware implementations and has a balanced Feistel structure. To increase the flexibility SIMON has been designed with key size 64, 72, 96, 128, 144192 or 256 and block size of 32, 48, 64, 96, or 128. It uses 32,36 42, 44, 52, 54, 68, 69 or 72 rounds depending on the key size and block size [93, 94].
9. SPECK: This cipher was also proposed by the National Security Agency (NSA) to optimize the performances in software implementations. It has a block size of 32, 48, 64, 96, or 128bits and uses a key size of 64, 72, 96, 128, 144, 192, or 256 with 22-34 rounds depending on key size and block size [95].
10. QTL: QTL is a generalized Feistel structure that has a 64-bit block size and 64 or 128-bit key size [96].
11. LEA: Lightweight Encryption Algorithm or simply LEA is a 128-bit block cipher with the key size of 128, 192, and 256-bits and uses 24, 28, and 32 rounds respectively [97]. Dungeon Lee et. Al. [98] reported on the hardware implementation of block cipher LEA. The design was applied in Verilog HDL, then incorporated into an FPGA chip and ASCI. Implementation results were compared with other results and it was observed that in throughput per area results were not up to mark, nevertheless, it is high in throughput.
12. SEA: Scalable Encryption Algorithm or simply SEA has a small code size with a limited set of instructions. For this reason, the SEA is used for small encryption devices. It has a Feistel structure [99].

Light-Weight CryptoGraphic	*Structure*	*Rounds*	*Key Size*	*Block Size*	*Weakness*
Tea	Feistel	64	128	-	*Bad as hash function, related key attack*
XTea	Feistel	64	128	64	*Related key rectangular attack on 36 round*
AES	SPN	10,12,14	128,192,256	128	*Bi-clique cryptanalysis*
Present	SPN	31	80\128	64	*Side channel attack, related key attack on 17 round*
DESL	Feistel	16	56	64	-
CLEFIA	Feistel	2488	128,192,256	39\128	*Differential fault analysis*
KATAN & KATANTAN	NLFSR/NLFSR	254&254	80&80	32\48\64	*Multidimensional meet in middle*
Humming_Bird	SPN	4	256	16	*Several attacks*
LED	SPN	8,12	64\80\96\128	64	*Bi-clique attack on reduced rounds*
Twine	Feistel	32	80\128	64	*Meet in the middle attack*
Klein	SPN	12\16\20	64,80,96	64	*Truncated differential attack*
Prince	SPN	11	128	64	*Es is questionable, choice for new attack*
Simon	Feistel	32-72	64-256	32-128	*Attack on reduced version and differential fault analysis*
QTL	Feistel	16\20	64\128	64	*Susceptible to linear differential attack*
Speck	Feistel	32-72	64-256	32-128	*Attack on reduced version and differential fault analysis*
LEA	Feistel	24,28,32	128,192,256	128	-
SEA	Feistel	-	-	-	-
Hight	Feistel	32	128	64	*Impossible differential attack on 26th round*
Print	SPN	48\96	80\160	48\96	-
GIFT	SPN	28\40	128	64\128	-

IV. Conclusion:

Here in this paper, the comparative performance of some lightweight cryptographic algorithms is presented. Their keysize, the number of rounds, and possible attacks have been mentioned but with one metric designer often need to trade off other important optimization goals like cost, energy, memory, performance, etc. So, the further extension of this research would be to customize a new cryptographic primitive and increase the overall performance.

References

[1]. S. Al Salami, J. Baek, K. Salah and E. Damiani, "Lightweight Encryption for Smart Home," 2016 11th International Conference on Availability, Reliability and Security (ARES), Salzburg, 2016, pp. 382-388.

[2]. A. Beg, T. Al-Kharobi and A. Al-Nasser, "Performance Evaluation and Review of Lightweight Cryptography in an Internet-of-Things Environment," 2019 2nd International Conference on Computer Applications & Information Security (ICCAIS), Riyadh, Saudi Arabia, 2019, pp. 1-6.

[3]. I. Bhardwaj, A. Kumar and M. Bansal, "A review on lightweight cryptography algorithms for data security and authentication in IoTs," 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC), Solan, 2017, pp. 504-509

[4]. Surendran, S., Nassef, A. and Beheshti, B., 2018. A survey of cryptographic algorithms for IoT devices. 2018 IEEE Long Island Systems, Applications and Technology Conference (LISAT),

[5]. I. K. Dutta, B. Ghosh and M. Bayoumi, "Lightweight Cryptography for Internet of Insecure Things: A Survey," 2019 IEEE 9th Annual Computing and Communication Workshop and Conference (CCWC), Las Vegas, NV, USA, 2019, pp. 0475-0481, doi: 10.1109/CCWC.2019.8666557.

[6]. Al-ahdal, Abdulrazzaq&Deshmukh, Nilesh. (2020). A Systematic Technical Survey Of Lightweight Cryptography On Iot Environment. International Journal of Scientific & Technology Research.

[7]. S. Sallam and B. D. Beheshti, "A Survey on Lightweight Cryptographic Algorithms," TENCON 2018 - 2018 IEEE Region 10 Conference, Jeju, Korea (South), 2018, pp. 1784-1789.

[8]. Dhanda, S.S., Singh, B. & Jindal, P. Lightweight Cryptography: A Solution to Secure IoT. Wireless PersCommun 112, 1947–1980 (2020).

[9]. Shah A., Engineer M. (2019) A Survey of Lightweight Cryptographic Algorithms for IoT-Based Applications. In: Tiwari S., Trivedi M., Mishra K., Misra A., Kumar K. (eds) Smart Innovations in Communication and Computational Sciences. Advances in Intelligent Systems and Computing, vol 851.Springer, Singapore.

[10]. Singh, S., Sharma, P.K., Moon, S.Y. et al. Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. J Ambient Intell Human Comput (2017).

[11]. M. A. Philip and Vaithiyanathan, "A survey on lightweight ciphers for IoT devices," 2017 International Conference on Technological Advancements in Power and Energy (TAP Energy), Kollam, 2017, pp. 1-4.

[12]. Hatzivasilis, G., Fysarakis, K., Papaefstathiou, I. et al. A review of lightweight block ciphers. J CryptogrEng 8, 141–184 (2018).

[13]. Sadkhan, S. and Salman, A., 2018. A survey on lightweight-cryptography status and future challenges. 2018 International Conference on Advance of Sustainable Engineering and its Application (ICASEA), Wasit, 2018, pp. 105-108.

[14]. Ray, S., 2020. Cryptographic Hashing | Hacker Noon. [online] Hackernoon.com. Available. at: https://hackernoon.com/cryptographic-hashing-c25da23609c3

[15]. Biryukov, A., Shamir, A., & Wagner, D. (2001). Real time cryptanalysis of A5, 1 on a PC, Fast Software Encryption (FSE), LNCS (Vol. 1978, pp. 1–18). New York: Springer.

[16]. Boesgaard, M., Vesterager, M., Pedersen, T., Christiansenm, J., &Scavenius, O. (2003). Rabbit: A new high-performance stream cipher, FSE, LNCS (Vol. 2887, pp. 307–329). Lund: Springer.

[17]. Ecrypt.eu.org. 2020. The Estream Portfolio Page. [online] Available at: https://www.ecrypt.eu.org/stream/

[18]. Bernstein D.J..The Salsa20 family of stream ciphers, CR. YP. TO, 2007. (Available from: http://cr.yp.to/snuffle/salsafamily-20071225.pdf)

[19]. Boesgaard M, Vesterager M, Pedersen T, Christiansen J, Scavenius O. Rabbit: a new high-performance stream cipher, FSE 2003, LNCS, vol. 2887, Springer, Lund, Sweden, 2003; 307 329.

[20]. Wu H. The stream cipher HC-128, New Stream CipherDesigns The eSTREAM Finalists, LNCS, vol. 4986Springer, 2008; 39–47.

[21]. Berbain C, Billet O, Canteaut, A, et al. SOSEMANUK, a fast software-oriented stream cipher, NewStream Cipher Designs, LNCS, vol. 4986 Springer, 2008; 98–118.

[22]. Hell M, Johansson T, Meier W. Grain a stream cipher for constrained environments. International Journalof Wireless and Mobile Computing 2007; 2 (1/2007): 86–93.

[23]. Dai W. Crypto++ Library 5.6.2, Crypto++, 2013. (Available from: http://www.cryptopp.com/)

[24]. Babbage S, Dodd M. The Stream Cipher MICKEY2.0, eStream Project, 2006. (Available from: http://www.ecrypt.eu.org/stream/p3ciphers/mickey/mickey_p3.pdf)

[25]. Boesgaard M, Vesterager M, Christiansen J, Zenner E. The Stream Cipher Rabbit 1, eStreamProject,2007.(Availablefrom:http://www.ecrypt.eu.org/stream/p3ciphers/rabbit/rabbit_p3.pdf)

[26]. Wu H. A new stream cipher HC-256, FSE 2004, LNCS, vol. 3017, Springer, Delhi, India, 2004;226–244.

[27]. Good T, Benaissa M. Hardware results for selected stream cipher candidates, SASC 2007, Bochum, Germany, 2007; 191–204.

[28]. Kircanski A, Youssef A M. Differential fault analysis of HC-128, AFRICACRYPT 2010, LNCS, vol. 6055, Springer, Stellenbosch, South Africa, 2010; 261–278.

[29]. Stankovski P, Ruj S, Hell M, Johansson T. Improved distinguishers for HC-128. Design, Codes and Cryptography 2012; 63 (2): 225–240, Springer.

[30]. Ekdahl P, Johansson T. A new version of the stream cipher snow, SAC 2002, LNCS, vol. 2595, Springer, Newfoundland, Canada, 2003; 47–61.

[31]. Anderson R, Biham E, Knudsen L. Serpent: A Proposal for the Advanced Encryption Standard, AEScontest,1998.(Available from: http://www.cl.cam.ac.uk/~rja14/Papers/serpent.pdf)

[32]. Berbain C, Billet O, Canteaut, A, et al. SOSEMANUK, a fast software-oriented stream cipher, NewStream Cipher Designs, LNCS, vol. 4986 Springer, 2008; 98–118.

[33]. Good T, Benaissa M. Hardware results for selected stream cipher candidates, SASC 2007, Bochum, Germany, 2007; 191–204

[34]. Ma Z, Gu D. Improved differential fault analysis of SOSEMANUK, 8th International Conference onComputational Intelligence and Security (CIS), IEEE, Guangzhou, China, 2012; 487–491

[35]. Hell M, Johansson T, Meier W. Grain a stream cipher for constrained environments. International Journalof Wireless and Mobile Computing 2007; 2 (1/2007): 86–93.

[36]. Manifavas, C., Hatzivasilis, G., Fysarakis, K. and Papaefstathiou, Y., 2015. A survey of lightweight stream ciphers for embedded systems. Security and Communication Networks, 9 (10), pp.1226-1246.

[37]. Mentens N, Genoe J, Preneel B, Verbauwhede I. A low-cost implementation of trivium, SASC 2008, Lausanne, Switzerland, 2008; 197–204.

[38]. Lu Y, Vaudenay S. Faster correlation attack on Bluetooth keystream generator E0, Advances in CryptologyCRYPTO 2004, LNCS, vol. 3152, Springer, Santa Barbara, California, USA, 2004; 407–425.

[39]. Mohamed MSE, Bulygin S, Buchmann J. Improved differential fault analysis of trivium, COSADE 2011, Darmstadt, Germany, 2011; 147–158.

[40]. Good T, Benaissa M. Hardware performance of estream phase-III stream cipher candidates, SASC2008, Lausanne, Switzerland, 2008; 163–174.

[41]. Banik S, Maitra S, Sarkar S. Improved differential fault attack on MICKEY 2.0. Report 2013/029, Cryptology ePrint Archive, IACR, 2013.

[42]. Babbage S, Dodd M. The Stream Cipher MICKEY-128 2.0, EsTram Project, 2006. (Available from: http://www.ecrypt.eu.org/stream/p2ciphers/mickey128/mickey128_p2.pdf)

[43]. Ding L, Guan J. Cryptanalysis of MICKEY family of stream ciphers. Security and Communication Networks 2013; 6 (8): 936–941.

[44]. Watanabe, D., Ideguchi, K., Kitahara, J., Muto, K., &Furuichi, H. (2008). Enocoro-80: A hardware oriented stream cipher. In Third International Conference on Availability, Reliability and Security(ARES 08) 2008; 1294 (1300): 4–7.49. Systems Development Laboratory, Hitachi.

[45]. Tahir R, Javed Y, Cheema AR. Rabbit-MAC: lightweight authenticated encryption in wireless sensor networks, IEEE International Conference on Informationand Automation, Zhangjiajie, China, 2008; 573–577.

[46]. Kumar N, Ojha S, Jain K, Lal S. BEAN: a lightweight stream cipher, 2nd International Conference onSecurity of Information and Networks (SIN '09), Gazimagusa, North Cyprus, 2009; 168–171.

[47]. Engels D, Fan X, Gong G, Hu H, Smith EM. Hummingbird: ultra-lightweight cryptography for resourceconstrained devices, Financial Cryptography and Data Security - FC 2010, LNCS, vol. 6054, Springer, Tenerife, Canary Islands, Spain, 2010; 3–18.

[48]. Chen T, Ge L, Wang X, Jiamei C. TinyStream: a lightweight and novel stream cipher scheme for wireless sensor networks, CIS 2010, Nanning, China, 2010; 528–532.

[49]. Engels D, Saarinen MJO, Schweitzer P, Smith EM. The hummingbird-2 lightweight authenticated encryption algorithm, RFID Sec 2011, Amherst,Massachusetts, USA, 2011; 19–31.

[50]. Tian Y, Chen G, Li J. Quavium - a new stream cipher inspired by trivium. Journal of Computers 2012; 7 (5): 1278–1283.

[51]. Sandip K, DebdeepM, Roy CD. Cavium strengthening trivium stream cipher using cellular automata.Journalof Cellular Automata 2012; 7 (2): 179–197.

[52]. Das S, Chowdhury RD. CAR30: a new scalable stream cipher with rule 30. Cryptography and Communications2013; 5 (2): 137–162.

[53]. Bogdanov A, Mendel F, Regazzoni F, Rijmen V, Tischhauser E. ALE: AES-based lightweight authenticated encryption, FSE'13, LNCS, Springer, Singapore, 2013.

[54]. Wu H. ACORN: A Lightweight Authenticated Cipher, CAESAR competition, 2014. (Available from: http://competitions.cr.yp.to/round1/acornv1.pdf)

[55]. Armknecht, F., &Mikhalev, V. (2015) On lightweight stream ciphers with shorter internal states. In G. Leander (Ed.),Fast Software Encryption: 22nd International Workshop, FSE 2015, Istanbul,Turkey, March 8–11, 2015, Revised Selected Papers (pp. 451–470). Berlin: Springer.https://doi.org/10.1007/978-3-662-48116 -522.

[56]. Ghafari, V. A., Hu, H., Xie, C. (2016). Fruit V2: Ultra-lightweight Stream Cipher with Shorter Internal State. Cryptology ePrint Archive Report 2016/355.http://eprin t.iacr.org/2016/355.

[57]. Mikhalev, V., Armknecht, F., & Muller, C. (2017). On ciphers that continuously access the non-volatile key. IACR Transmission Symmetric Cryptology, 2, 52–79. https://doi.org/10.13154 /tosc.v2016. i2.52-79.

[58]. Dubrova, E., & Hell, M. (2017). Espresso: A stream cipher for 5G wireless communication systems. Journal of Cryptography and Communication, 9 (2), 273–289.

[59]. Hamann, M., Krause, M., & Meier, W. (2017). LIZARD—A lightweight stream cipher for powerconstrained devices. IACR Transmission Symmetric Cryptology, 1, 45–79. https://doi.org/10.13154 /tosc.v2017. i1.45-79.

[60]. Wikipedia. 2020. Cryptographic Hash Function. [online] Available at: https://simple.wikipedia.org/wiki/Cryptographic_hash_function.

[61]. Z. Al-Odat and S. Khan, "The sponge structure modulation application to overcome the security breaches for the md5 and sha-1 hash functions," in 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), vol. 1, Jul 2019, pp. 811–816.

[62]. Aumasson, J.-P., Henzen, L., Meier, W., &Naya-Plasencia, M. (2010). Quark: A lightweight hash. InInternational Workshop on Cryptographic Hardware and Embedded Systems (pp. 1–15).Springer.

[63]. Hirose, S., Ideguchi, K., Kuwakado, H., Owada, T., Preneel, B., & Yoshida, H. (2010). A lightweight256-bit hash function for hardware and low-end devices: Lesamnta-LW. In Proceeding of InternationalConference on Information Security and Cryptology (pp. 151–168). Berlin: Springer.

[64]. Kavun, E. B., &Yalcin, T. (2010). A lightweight implementation of keccak hash function for radiofrequency identification applications. In International Workshop on Radio Frequency Identification: Security and Privacy Issues (pp. 258–269). Springer.

[65]. Guo, J., Peyrin, T., &Poschmann, A. (2011). The PHOTON family of lightweight hash functions, CRYPTO 2011, LNCS 6841, International Association for Cryptologic Research (pp. 222–239).

[66]. Bogdanov, A., Kneˇzevi´c, M., Leander, G., Toz1, D., Varıcı, K, &Verbauwhede, I. (2011). SPONGENT: A lightweight hash function, CHES 2011, LNCS 6917, International Association for Cryptologic Research (pp. 312–325)

[67]. Berger, T. P., D'Hayer, J., Marquet, K., Minier, M., & Thomas, G. (2012). The GLUON family: A lightweight hash function family based on FCSRs. In A. Mitrokotsa& S. Vaudenay (Eds.) Progress in Cryptology—AFRICACRYPT 2012. Lecture Notes in Computer Science, Vol. 7374. Springer, Berlin.

[68]. Arnault, F., Berger, T., Lauradoux, C., Minier, M., Pousse, B.: A New Approach for FCSRs. In: Jacobson Jr., M.J., Rijmen, V., Safavi-Naini, R. (eds.) SAC 2009. LNCS, vol. 5867, pp. 433–448. Springer, Heidelberg (2009)

[69]. Berger, T.P., Minier, M., Pousse, B.: Software Oriented Stream Ciphers Based upon FCSRs in Diversified Mode. In: Roy, B., Sendrier, N. (eds.) INDOCRYPT 2009. LNCS, vol. 5922, pp. 119–135. Springer, Heidelberg (2009)

[70]. Wenling, W., Shuang, W., Zhang, L., Zou, J., & Dong, L. (2013). LHash: A lightweight hash function (full version). https://eprin t.iacr.org/2013/867.

[71]. Mukundan, P. M., Manayankath, S., Srinivasan, C., &Sethumadhavan, M. (2016). Hash-One: A lightweight cryptographic hash function. IET Information Security, 10 (5), 225–231.

[72]. Bussi, K., Dey, D., Kumar, M., &Dass, B. K. (2016) Neeva: A Lightweight Hash Function, IACR Cryptology ePrint Archive, (042). https://eprin t.iacr.org/2016/042.

[73]. National Institute of Standards and Technology (NIST). (2001). Advanced Encryption Standard (AES).Federal information processing standards publication 197, November 26.http://csrc.nist.gov/publicatio ns/fips/fips1 97/fips-197.pdf.

[74]. Bogdanov, A., Knudsen, L. R., Leander, G., Paar, C., Poschmann, A., Robshaw, M. J. B., Seurin, Y., Vikkelsoe, C.: PRESENT: An ultra-lightweight block cipher. In Proceeding of CryptographicHardware and Embedded Systems—CHES 2007 (pp. 450–466). Springer.

[75]. Engels D., Fan X., Gong G., Hu H., Smith EM (2010) Hummingbird: Ultra-Lightweight Cryptography for Resource-Constrained Devices. In: Sion R. et al. (eds) Financial Cryptography and Data Security. FC 2010. Lecture Notes in Computer Science, vol 6054. Springer, Berlin, Heidelberg.

[76]. Anjali Arora, Priyanka, Saibal Kumar Pal, "A Survey of Cryptanalytic Attacks on Lightweight Block Ciphers," IRACST - International Journal of Computer Science and Information Technology & Security (IJCSITS), ISSN: 2249-9555 Vol. 2, No.2, April 2012

[77]. Knudsen L., Leander G., Poschmann A., Robshaw M.J.B. (2010) PRINTcipher: A Block Cipher for IC-Printing. In: Mangard S., Standaert FX. (eds) Cryptographic Hardware and Embedded Systems, CHES 2010.CHES 2010. Lecture Notes in Computer Science, vol 6225. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-15031-9_2

[78]. Zheng Gong, SvetlaNikova and YeeWeiLaw,KLEIN: A New Family of Lightweight Block Ciphers, A. Juels and C. Paar (Eds.): RFIDSec 2011, LNCS 7055, pp. 1–18, 2012.

[79]. Guo J., Peyrin T., Poschmann A., Robshaw M. (2011) The LED Block Cipher. In: Preneel B., Takagi T. (eds) Cryptographic Hardware and Embedded Systems–CHES 2011. CHES 2011. Lecture Notes in Computer Science, vol 6917. Springer, Berlin, Heidelberg.

[80]. Borghoff J. et al. (2012) PRINCE–A Low-Latency Block Cipher for Pervasive Computing Applications. In: Wang X., Sako K. (eds) Advances in Cryptology–ASIACRYPT 2012. ASIACRYPT 2012. Lecture Notes in Computer Science, vol 7658. Springer, Berlin, Heidelberg.

[81]. Morawiecki, P., 2017. Practical attacks on the round-reduced PRINCE. IET Information Security, 11 (3), pp.146-151.

[82]. SubhadeepBanik, Sumit Kumar Pandey, Thomas Peyrin, Yu Sasaki, Siang MengSim, and Yosuke Todo. GIFT: A small present - towards reaching the limit of lightweight encryption. In Cryptographic Hardware and Embedded Systems - CHES 2017 - 19th International Conference, Taipei, Taiwan, September 25-28, 2017, Proceedings, pages 321–345, 2017.

[83]. Hong D. et al. (2006) HIGHT: A New Block Cipher Suitable for Low-Resource Device. In: Goubin L., Matsui M. (eds) Cryptographic Hardware and Embedded Systems - CHES 2006. CHES 2006. Lecture Notes in Computer Science, vol 4249. Springer, Berlin, Heidelberg.

[84]. Leander, Gregor&Paar, Christof&Poschmann, Axel & Schramm, Kai. (2007). New lightweight DES variants. Lect. Note. Comput.Sci.. 4593. 196-210. 10.1007/978-3-540-74619-5_13.

[85]. M. B. Abdelhalim, M. El-Mahallawy, M. Ayyad and A. Elhennawy, "Implementation of a modified lightweight cryptographic TEA algorithm in RFID system," 2011 International Conference for Internet Technology and Secured Transactions, Abu Dhabi, 2011, pp. 509-513.

[86]. Suzaki, T., Minematsu, K., Morioka, S., & Kobayashi, E. (2011) TWINE: A lightweight, versatile block cipher. In Proceeding of ECRYPT Workshop on Lightweight Cryptography 2011 (pp. 146–169).

[87]. De Canniere, C., Dunkelman, O., &Kneževi´c, M. (2009). KATAN and KTANTAN—a family of small and efcient hardware-oriented block ciphers. In International Workshop on Cryptographic Hardware and Embedded Systems (pp. 272–288).Springer.

[88]. Shirai T., Shibutani K., Akshita T., Moriai S., Iwata T. (2007) The 128-Bit Blockcipher CLEFIA (Extended Abstract). In: Biryukov A. (eds) Fast Software Encryption. FSE 2007. Lecture Notes in Computer Science, vol 4593. Springer, Berlin, Heidelberg.

[89]. Minier, M. and Naya-Plasencia, M., 2012. A related key impossible differential attack against 22 rounds of the lightweight block cipher LBlock. Information Processing Letters, 112 (16), pp.624-629.

[90]. M. N. Hasan, M. T. Hasan, R. N. Toma and M. Maniruzzaman, "FPGA implementation of LBlock lightweight block cipher," 2016 3rd International Conference on Electrical Engineering and Information Communication Technology (ICEEICT), Dhaka, 2016, pp. 1-4, doi: 10.1109/CEEICT.2016.7873062.

[91]. En.wikipedia.org. 2020. Simon (Cipher). [online] Available at: https://en.wikipedia.org/wiki/Simon_(cipher)

[92]. S. Feizi, A. Ahmadi and A. Nemati, "A hardware implementation of Simon cryptography algorithm," 2014 4th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, 2014, pp. 245-250,

[93]. En. Wikipedia.org. 2020. Speck (Cipher). [online] Available at: https://en.wikipedia.org/wiki/Speck_(cipher)

[94]. Li, L., Liu, B. and Wang, H., 2016. QTL: A new ultra-lightweight block cipher. Microprocessors and Microsystems, 45, pp.45-55.

[95]. En. Wikipedia.org. 2020. LEA (Cipher). [online] Available at: https://en.wikipedia.org/wiki/LEA_(cipher).

[96]. Lee, D., Kim, D., Kwon, D. and Kim, H., 2014. Efficient Hardware Implementation of the Lightweight Block Encryption Algorithm LEA. Sensors, 14 (1), pp.975-994.

[97]. Standaert, François-Xavier &Piret, Gilles &Gershenfeld, Neil &Quisquater, Jean-Jacques. (2006). SEA: A Scalable Encryption Algorithm for Small Embedded Applications. 222-236. 10.1007/11733447_16.