A performance comparison of lightweight cryptographic algorithms suitable for IoT transmissions

Jihane Jebrane^{a,*}, Saiida Lazaar^{a,*}

^aMathematics, Computer Science and Applications ERMIA-Team. ENSA of Tangier, AbdelMalek Essaadi University. Morocco.

Abstract

In the past decade, the Internet of Things (IoT) has emerged and has been of great importance in the digital world arousing the interest of the scientific community. The requirements of new small intelligent devices brought numerous security and privacy concerns. In order to secure the exchanged data through the IoT, integrity and authentication protocols are required and are usually constructed under hash functions. We notice that the most conventional cryptographic standards are not suitable for multiple platforms hardware/software. The expressed need is to construct lightweight functions and algorithms suitable for constrained environments including small devices. In this context, we propose a review of the research progress on IoT architecture. Further, we present and compare some selected lightweight cryptographic algorithms regarding security, performance, and resource requirements. The main goal of the present paper is to highlight the need to provide security systems for IoT applications through lightweight cryptographic mechanisms.

Keywords: Internet of Things, Lightweight Cryptography, Block Cipher, Hash function, Stream Cipher.

1. Introduction

The Internet of Things (IoT) is the latest internet evolution that enables many low resources and constrained devices to communicate with each other. It means that everything real becomes a virtual representation on IoT. IoT is a developing term that combines three concepts; Things are locatable, scalable, and heterogeneous [1]. In 2013, Gartner points out that IoT will generate more than \$300 billion in revenue until 2020. Furthermore, the number of smartphones and tablets will reach up to 7.3 billion units by 2020 [2]. As IoT is growing rapidly, this exponential evolution will explode a massive amount of data communicated throughout the network. Besides, if these data are compromised, the whole system might suffer seriously. Therefore, security plays an important role in the network to protect data against unauthorized access, misuses, monitoring, and any modifications, etc. In recent years, the majority of cryptographic algorithms standards designed to fit into the desktop/server environment but not intended to be implemented on constrained devices due to the limitation of their resources. For passive RFID tags, they do not have battery-powered onboard, but instead, they absorb their power from the environment (RFID-Reader). Therefore, these constrained devices require efficient cryptographic functions that meet a very small amount of gate equivalents (GEs), stringent timing and energy consumption. For these reasons, the limitations of conventional cryptographic algorithms in resource-constrained devices

*Jihane Jebrane

Email addresses: jihanejbr1@gmail.com (Jihane Jebrane), slazaar@uae.ac.ma (Saiida Lazaar)

doi:[10.31559/glm2021.10.2.5](https://doi.org/10.31559/glm2021.10.2.5)

Received 30 Apr 2021 : Revised : 25 May 2021 Accepted: 25 Jun 2021

impose many challenges to developing a new subfield of cryptography, which fulfills an optimist security mechanism needed. This new research area is referred to as lightweight cryptography (LWC). According to NIST lightweight cryptography is a new branch of cryptography that is developed to delivers adequate secure algorithms for constrained devices and aims to reduce the key size, cycle rate, throughput rate, power consumption, and area. Lightweight cryptography is a very large field which should be divided into two different areas: Ultra-lightweight and IoT cryptography [3]. An ultra-lightweight cryptographic algorithm is tailored to working on very cheap devices that have a limited shelf-life and that not connected to the internet. Such as RFID tags, smart cards, remote car keys, memory encryption and so on. Note that an ideal ultra-lightweight algorithm should have at least 80 bits in key size, 64 bits (or more) in block size, and a block /stream cipher as a type [3]. Whereas an IoT cryptographic algorithm is designed to running on low-resource devices interconnected to a public channel such as the internet. For that reason, the security mechanism must be more stringent than in the ultra-lightweight case. Moreover, its implementation should have 96 bits in block size, at least 128 bits in key size, and block cipher or sponge as a type [4]. The aforementioned comparison is therefore to understand the need for lightweight algorithms in order to provide new conservatives solutions that can be implemented over resource-limited devices. This paper provides an overview of IoT architecture in section 2. Next, we highlight and compare of different primitives of lightweight cryptographic algorithms based on their behavior on a performed platform in section 3. Lastly, we close the paper with a conclusion and references.

2. Overview of IoT architecture

IoT connects billions of things called objects (sensors, actuators, embedded devices, traditional computers, smartphones). These objects will establish a huge complex network for sharing information. Thus, IoT should take multiple decisions in real-time to manage the interconnection between these various objects. This is accomplished over the architecture of IoT. In July 2014 in Germany, the first meeting of the IEEE P2413 working group has shown that the standard architecture for IoT is still in progress [5]. Moreover, the conventional internet protocol cannot support M2M communication. In this section, we discuss several architectures of IoT that have been proposed in the literature. According to [1, 6, 7, 8, 9, 10] the commonly IoT architectures are divided into three layers architecture; Perception or physical layer, network or commutation layer and application layer. The first layer (also known as the perceptual layer) is responsible for information acquisition for each object which helps in collecting the information in the IoT environment such as sensors, meters, RFID systems, and GPS systems, etc. The second layer is the network layer it is also known as the core of IoT consists to transmit information collected by the physical layer through the public network. This layer includes all transmission media such as WSNs, 5G, 4G, 3G, 2G, Internet, fiberoptic. Finally, the application layer defines as a method to identify the mold of application that will be used in IoT. The three-layer architecture presents general information but does not give numerous details about IoT structure and management of data congregate [6]. Authors in [7] added a new layer between the network and the application layer known as data management. This layer is responsible to manage data transmission and storage by centers for cloud computing and directory management servers. The four-layer architecture of IoT considers that it is the most suitable model. Authors in [6, 10, 11] proposed five layers by adding two layers which are the middleware or processing layer and the business layer. The first one positioned between the network and application layer in order to manage and analyze data congregated by the physical layer. The second one lays above the application layer and is responsible to present the design of the IoT applications and their management. Figure 1 summarizes of several architectures in IoT. Along with this, the most important criteria that should have in IoT architecture is the security guarantee. Therefore, the architecture should be able to protect data coming and going over the public network, to provide the quality of service, to endure passive and active attacks, and to guarantee the confidentiality and integrity protection of information sharing.

Figure 1: Summary of existing IoT architecture.

3. Related work

This paper outlines recent technologies of lightweight cryptographic primitives used on performed platforms. These primitives offer several advantages over traditional cryptographic standards. There are two main reasons to adopt new particular technology for resource-constrained environments. Firstly, the efficiency of the lightweight symmetric key algorithm to realize end-to-end communications security with limited energy consumption. On the other hand, the footprints of lightweight algorithms are smaller than the classical cryptographic ones. The lightweight cryptography primitives are capable to create more network connections in the low resource devices [4, 12]. The lightweight cryptographic algorithms are divided into two types, symmetric and asymmetric ciphers as described below.

3.1. Symmetric Lightweight algorithms for IoT

This kind of encryption pre-shared the same key for both encryption and decryption of data. The symmetric key encryption considered secure and fast but does not guarantee authentication because if the key's compromised, then the encrypted data gets extremely attacked. Symmetric ciphers include block, stream, and hash-functions. We report many different technologies of lightweight cryptography primitives.

3.1.1. Lightweight Block ciphers

In the last decade, there are a large number of methods which have been proposed and are currently used in lightweight block ciphers in order to achieve the best performance advantages in low resource smart devices such as Advanced Encryption Standard (AES-128) [13], which is based on substitution permutation network (SPN), it supports 128 bits block length, works with different key sizes of 128 with 10 rounds, 192 bits/ 12 rounds and 256 bits /14 rounds. Next is, DESL [14]. Or DES Light Weighted.

Basically, this cipher is proposed by simplifying the classical DES in order to enhance their efficiency. The DESL round function works with the single S-box instead of eight, which decreases hardware gate complexity. In [22] the comparison results between DESL with DES, DESX, DESXL, and also AES show that DESL is much appropriate for RFID tags by giving the smallest (35%) gate equivalence. PRESENT [15] is also based on SPN structure, it has a key size of 80 or 128 bits with 31 rounds, and uses on block of 64 bit. The PRESENT algorithm considered one of the first ultra-lightweight block ciphers that were designed from scratch in order to guarantee efficient security for a constrained hardware environment, but for software implementation, the substitution layer that applies 4-bits of input and the S-box output consumes large cycles. For that reason, an improved version was proposed known as RECTANGLE cipher [6]. RECTANGLE [16] and PRESENT are quite similar. The first one consists of just 25 rounds and considered more performed and suitable for both platform's hardware and software implementations with limited resources.

Table 1: Summary of some lightweight block cipher.

Ref	Algorithm	Structure	Block size	Key size	Rounds	Application	Attacks
13	AES	SPN	128	128/192/256	10/12/14	Not given	Related key attack Boomerang attack Biclique cryptanalysis.
14	DESL	FN	64	54	16	RFID	differential attack on a second round with probability 1/234.
15	PRESENT	SPN	64	80/128	32	RFID tags	Bottleneck attacks truncated differential Side-channel attacks.
16	RECTANGLE	SPN	64	128	26	Not given	slide attack related-key Saturation Attack.
17	CLEFIA	FN	128	128	18	Not given	Key Recovery Attack on 10th round Saturation Cryptanalysis
19	HIGHT	FN	64	128	32	RFID(FPGA)	Impossible differential attack on 26th round Biclique cryptanalysis.
20	CAMELIA	SPN	128	128	18/24	Not given	timing attacks Impossible differential
21	TEA	FN	64	128	64	Energy server	Related key attack Differential fault
23	SIMON	SPN	128	128	64	ASIC	Attacks on reduced versions
24	SPECK	SPN	128	128	32	Not given	Key Recovery Boomerang attack
25	TWINE	FN	64	80/128	36	Not given	Meet-in-the middle Saturation Attack

CLEFIA [17] is based on Feistel network, it uses a block size of 128 bits and key lengths of 128, 192 and 256 bits and can be implemented with 6K gates [18]. Also, it has significant hardware performance in comparison to other block ciphers. HIGHT [19] (High Security and Lightweight) is based on Feistel network structure, it uses basic operations like addition mod 28 or XOR. It has a block size of 64-bits and works with 128-bit key, with 32 rounds [1]. CAMELLIA [20] is worked with a block size of 128 bits and 128,

192 and 256- bits of keys respectively. It was designed for both software and hardware implementations. The next is TEA [21] (Wheeler and Needham (1994)) and XTEA Tiny encryption algorithm is a block cipher based on ARX (Addition/Rotation/XOR) design, which consists of simple round structures that make them appropriate for constrained software environments. Later SIMON [24] and SPECK [25] were introduced in June 2013. SIMON is a lightweight cipher which is optimized for performance in hardware implementations, whereas SPECK has been dedicated for software implementations, there are two new proposed primitives MARX and SPECKEY which contain certain limits against single-trail differential and linear cryptanalysis. According to [22], SIMON was compared to AES, PRESENT, SPECK, TWINE, and PRINCE, it was concluded that SIMON and SPECK are ideal for use through heterogeneous networks, they are efficient in implementation than AES and also very suitable to work with. The last algorithm is TWINE [26] is based on Feistel structure, and when it was compared to AES, PRESENT, HIGHT, and Piccolo. The comparison results show that it was designed to fit in both hardware and software implementations. TWINE has two types, TWINE and TWINE8 with their respective key sizes. Both algorithms have 64 bits block size and 36 rounds. The comparative analysis of the aforementioned lightweight block ciphers is explained in Table 1 [22].

3.1.2. Lightweight hash functions

The research on lightweight dedicated hash functions held from 2006, when Feldhofer and Rechberger evoked the lack of applying lightweight hash functions in the RFID tags [2]. The Lightweight hash function creates output messages with a fixed length from arbitrary input length messages. The output is known as message digest and is used to secure many applications like Message Authentication Code (MAC), pseudo number generator for passwords, digital signatures and the integrity of the transmitted data.

Figure 2: Hash function.

The hash function involves two main parts are construction and compression function. The design of the construction function is based on two main steps. The first step is considered to construct a compression function with a fixed domain.

$$f : \{0, 1\}^n \times \{0, 1\}^m \rightarrow \{0, 1\}^n \quad m, n > 0$$

The second step is to iterating this compression function many times to determine a hash function. This method is known as domain extension or design of iteration [23].

$$H : \{0, 1\}^* \rightarrow \{0, 1\}^n$$

An ideal hash function with an n-bit output required three major security properties are given below.

- Collision resistance : The pair of different messages (M_1, M_2) should be difficult to find such that $H(M_1) = H(M_2)$.
- Preimage resistance : Given a random output r , it is difficult to find a message M_1 such that $H(M_1) = r$.

- 2^{nd} Preimage resistance: Given a fixed size message M_1 , it is difficult to find M_2 with $H(M_1) = H(M_2)$.

Most conventional hash function algorithms required a huge internal state size, high energy consumption, and produce a large memory footprint, which is not suitable and efficient for resource-limited devices [3]. Furthermore, the implementation of lightweight hash functions is considered more complicated than lightweight block ciphers. Nevertheless, in 2007 Hirotaka Yoshida presents a new compression function known as MAME [22, 27] which is a lightweight cryptographic hash function. It takes a 256-bit message block and a 256-bit chaining variable as inputs and produces a 256-bit output. After that, the authors in [28] proposed a novel operation of a PRESENT block cipher in hashing modes. Thus, the National Institute of Standards and Technology (NIST) approved many lightweight hash algorithms which take 256 characters in input and produces a smaller memory footprint such as SPONGENT [29], PHOTON [30], QUARK [31], LESAMNTA-LW [32], and KECCAK [18]. In [22] the comparison results between PHOTON, KECCAK-200, KECCAK-400, and SPONGENT when implemented on Xilinx Spartan 6 FPGA, where SPONGENT can achieve a highest throughput/area ratio, and PHOTON showed more suitable scalability in the area but lowest throughput in the field.

3.1.3. Lightweight stream ciphers

Lightweight stream ciphers are also considered as efficient and promising for resource constrained IoT environment [4]. Compared with lightweight block algorithms, the stream cipher described as faster, easier, and flexible in hardware applications efficiency. It based on two constructions such as Linear Feedback Shift Registers (LFSRs) and Non-Linear Feedback Shift Registers (NLFSRs). Table 2 summarizes some lightweight stream ciphers in terms of their constructions, key size, gate area, and throughput.

Table 2: Comparative analysis of some lightweight stream ciphers.

Ref	Algorithm	Construction	Key size	Initialized-vector(IV)	Gate-area(GEs)	Throughput	attacks
32	WG8	LFSR	80	80	1984	High	Key recovery attack
33	GRAIN	Both	128	96	1300	Low	Differential fault attack
34	TRIVIUM	LFSR	80	80	2600	High	Improved differential fault attack
35	CHACHA	Both	256	128	750	High	Attacks on reduced versions
36	ESPRESSO	NLFSR	128	96	1500	High	IV attacks, linear approximation
37	MICKEY	LFSR	80	80	3188	Low	Improved differential fault attack

3.2. Asymmetric lightweight algorithms for IoT

In the case of asymmetric encryption, private and public keys are used for the establishment of communication between two entities. Usually, asymmetric encryption is applied to digital signatures (RSA, ECC) and key agreements (Diffie-Hellman (DH)) in order to provide the three noteworthy parts of

lightweight cryptography such as authentication, confidentiality, and integrity. Apart from this, asymmetric encryption should be able enough to support all features possible and guarantee a strong mechanism to convey keys between the sender and the receiver. However, these security mechanisms have a large key size which increases the computational complexities such as RSA by Rivest, Shamir, and Adleman, Diffie-Hellman key exchange (DH), and Elliptic Curve Cryptography (ECC). Consequently, the alternatives lightweight asymmetric algorithms used are identification scheme crypto GPS, Authentication, and Key exchange mechanism ALIKE [39] (which is a lightweight key exchange algorithm based on RSA with the apply of AES encryption), ID-based signature scheme IBS [4], and ELLI (Elliptic Light) [6].

Conclusion

Lightweight cryptography has received attention from several researchers in the past two decades. This paper reviews the most promising lightweight cryptography solutions over resource-limited devices in regard to memory, energy consumption, and computation complexities. In spite of rapid evolution, many lightweight mechanisms are still in their start. Furthermore, we summarized and compared different kinds of lightweight algorithms implemented on multiple platforms hardware and software.

References

- [1] E. R. Naru, H. Saini and M. Sharma *A recent review on lightweight cryptography in IoT*, 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, (2017), pp. 887-890. [1](#), [2](#), [3.1.1](#)
- [2] S. Singh, P.K. Sharma, S.Y. Moon *Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions.*, J Ambient Intell Human Comput, (2017). [1](#), [3.1.2](#)
- [3] A. Biryukov, L.P. Perrin *State of the art in lightweight symmetric cryptography*, 2017. [1](#), [3.1.2](#)
- [4] K. McKay, L. Bassham, M. Snmez Turan, N. Mouha *Report on lightweight cryptography*, 2016. [1](#), [3](#), [3.1.3](#), [3.2](#)
- [5] M. Saadeh, A. Sleit, M. Qatawneh, W. Almobaideen *Authentication Techniques for the Internet of Things: A Survey*, Cybersecurity and Cyberforensics Conference (CCC), Amman, 2016, pp. 28-34. [2](#)
- [6] I. Bhardwaj, A. Kumar, M. Bansal *A review on lightweight cryptography algorithms for data security and authentication in IoTs*, 4th International Conference on Signal Processing, Computing and Control (ISPCC), Solan, 2017, pp. 504-509. [2](#), [3.1.1](#), [3.2](#)
- [7] M. Zhang, F. Sun, X. Cheng *Architecture of Internet of Things and its Key Technology Integration Based-on RFID*, IEEE Fifth International Symposium on Computational Intelligence and Design, 2012, pp. 294-297. [2](#)
- [8] Z. Ma, X. Shang, X. Fu, F. Luo *The Architecture and Key Technologies of Internet of Things in Logistics*, 2013 International Conference on Cyberspace Technology (CCT), IET, Beijing, China 2013, pp. 464 - 468. [2](#)
- [9] M. Weyrich, C. Ebert *Reference Architectures for the Internet of Things*, IEEE SOFTWARE, 2016, vol.33, no.1, pp.112-116. [2](#)
- [10] S. Kraijak, P. Tuwanut *A Survey on Internet of Things Architecture, Protocols, Possible Applications, Security, Privacy, Real-World Implementation and Future Trends*, IEEE 16th International Conference on Communication Technology (ICCT), 2015, pp. 26-31. [2](#)
- [11] R. Khan, S. U. Khan, R. Zaheer, S. Khan *Future Internet: The Internet of Things Architecture, Possible Applications and Key Challenges*, IEEE 10th International Conference on Frontiers of Information Technology, 2012, pp. 257 ? 260. [2](#)
- [12] M. Katagi *Lightweight Cryptography for the Internet of Things*, 2011. [3](#)
- [13] k. Lokibe , K. Maeshima, H. Kagotanu, Y. Nogami, Y. Toyota, T. Watanabe *Analysis on equivalent current source of AES-128 circuit for HD power model verification.*, In Proceeding of International Symposium on Electromagnetic Compatibility. Tokto (EMC?14/Tokyo), IEEE, 2014, pp 302-305. [3.1.1](#)
- [14] G. Leander, C. Paar, A. Poschmann, K. Schramm *New lightweight DES variant*, In: Proceeding of International Workshop on Fast Software Encryption, Springer, Berlin, 2007, pp 196?210. [3.1.1](#)
- [15] A. Bogdanov, LR. Knudsen, G. Leander, C. Paar, A. Poschmann, J. RobshawM, C. Vikkelseo *PRESENT: an ultra-lightweight block cipher*, Proceeding of International Workshop on Cryptographic Hardware and Embedded Systems, Springer, Berlin, 2007, pp 450?466. [3.1.1](#)
- [16] W. Zhang *RECTANGLE: a bit-slice lightweight block cipher suitable for multiple platforms*, Science China Information Sciences, 2015, vol. 58(12), pp. 1-15. [3.1.1](#)
- [17] T. Akshita, H. Hiwatari *Very compact Hardware Implementations of the Block cipher CLEFIA*, Selected Areas in Cryptography Lecture Notes in Computer Science, Springer, 2012, pp. 278-292. [3.1.1](#)
- [18] W.J. Buchanan, Li. Shancang, R. Asif *Lightweight cryptography methods*, Journal of CyberSecurity Technology, 2017, 1:3-4, 187-201. [3.1.1](#), [3.1.2](#)
- [19] D. Hong *HIGHT: A New Block Cipher Suitable for Low Resource Device*, Cryptographic Hardware and Embedded Systems CHES 2006 Lecture Notes in Computer Science, 2006, pp. 46-59. [3.1.1](#)

- [20] A. Satoh, S. Morioka *Hardware-Focused Performance Comparison for the Standard Block Ciphers AES, Camellia, and Triple-DES*, Lecture Notes in Computer Science Information Security, Springer, 2003, pp. 252-266. [3.1.1](#)
- [21] DJ. Wheeler, RM. Needham *TEA, a tiny encryption algorithm*, Proceeding of International Workshop on Fast Software Encryption, Springer, Berlin, 1994, pp 363?36. [3.1.1](#)
- [22] A. Shah, M. Engineer *A Survey of Lightweight Cryptographic Algorithms for IoT-Based Applications.*, 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, 2019. [3.1.1](#), [3.1.1](#), [3.1.2](#)
- [23] R. Vidya, K.V. Prema *Light-weight hashing method for user authentication in Internet-of-Things*, Ad Hoc Networks, vol 89. Elsevier, 2019, pp 97-106. [3.1.2](#)
- [24] R. Beaulieu, D. Shors, J. Smith, S. Treatment-Clark, B. Weeks, L. Wingers *Simon and Speck: Block Ciphers for the Internet of Things*, NIST Lightweight Cryptography, 2015. [3.1.1](#)
- [25] A.V. Duka, B. Genge *Implementation of SIMON and SPECK Lightweight Block Ciphers on Programmable Logic Controllers*, IEEE, 2017. [3.1.1](#)
- [26] T. Suzuki, K. Minematsu, S. Morioka, E. Kobayashi *TWINE: A Lightweight Block Cipher for Multiple Platforms*, Springer, 2012. [3.1.1](#)
- [27] H. Yoshida *MAME: A Compression Function with Reduced Hardware Requirements*, Cryptographic Hardware and Embedded Systems - CHES, Berlin, Heidelberg, 2007, pp. 148?165. [3.1.2](#)
- [28] A. Bogdanov, G. Leander, C. Paar, A. Poschmann, MJ. Robshaw, Y. Seurin *Hash functions and RFID tags: mind the gap*, Proceeding of International Workshop on Cryptographic Hardware and Embedded Systems, Springer, Berlin, 2012, pp 283?299. [3.1.2](#)
- [29] A. Bogdanov, M. Kne?evic, G. Leander, D. Toz, K. Varici, I. Verbauwhede *SPONGENT: a lightweight hash function*, Proceeding of International Workshop on Cryptographic Hardware and Embedded Systems, Springer, Berlin, 2011, pp 312?325. [3.1.2](#)
- [30] J. Guo, T. Peyrin, A. Poschmann *The PHOTON family of lightweight hash functions*, Proceeding of Annual Cryptology Conference, Springer, Berlin, 2011, pp 222?239. [3.1.2](#)
- [31] JP. Aumasson, L. Henzen, W. Meier, M. Naya-Plasencia *Quark: a lightweight hash*, J Crypto, 2013, 26(2):313?339. [3.1.2](#)
- [32] S. Hirose, K. Ideguchi, H. Kuwakado, T. Owada, B. Preneel, H. Yoshida *A lightweight 256-bit hash function for hardware and low-end devices: lesamnta-LW*, Proceeding of International Conference on Information Security and Cryptology, Springer, Berlin, 2010, pp 151?168. [3.1.2](#)
- [33] X. Fan, K. Mandal, G. Gong *Wg-8: a lightweight stream cipher for resource-constrained smart devices*, Proceeding of International Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness, Springer, Berlin, 2013, pp 617?632.
- [34] M. Hell, T. Johansson, W. Meier, *Grain: a stream cipher for constrained environments*, Int J Wirel Mob Comput 2(1):86?93, 2007.
- [35] P. Kitsos, N. Sklavos, G. Provelengios, Skodras *AN FPGA based performance analysis of stream ciphers ZUC, Snow3g, Grain V1, Mickey V2, Trivium and E0*, Microprocess Microsyst, 2013, 37(2):235?245.
- [36] Bernstein, J. Daniel *The Salsa20 family of stream ciphers*, Lecture Notes in Computer Science, 2008, 4986:84-97.
- [37] E. Dubrova, M. Hell *Espresso: A stream cipher for 5G wireless communication systems*, Cryptogr. Commun, 2017, 9, 273?289.
- [38] S. Babbage, M. Dodd *The MICKEY stream ciphers*. In: *Proceeding of New Stream Cipher Designs*, Springer, Berlin, 2008, pp 191?209.
- [39] L. Ertaul, P. Chudinov, B. Morales *IoT Security: Authenticated Lightweight Key Exchange (ALIKE)*, 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, 2019, p. 6. [3.2](#)