Available online at www.refaad.com
Gen.Lett. Math., 10(2) (2021), 38-45
Research Article

General Letters in Mathematics (GLM)

Journal Homepage: http://www.refaad.com/views/GLM/home.aspx

A lightweight cryptographic solution to secure digital transmissions on resource-constrained environments

Saiida Lazaar[a,*]

[a] *Departement of Mathematics and Computer Science - ERMIA TEAM. National School of Applied Sciences - Tangier, ABDELMALEK ESSAADI University, Morocco*

Abstract

The great revolution of technology and its fast growth have led to a cyber space increasingly vulnerable to cyber-attacks. For this reason, cyber security becomes paramount to protect our cyber space by presenting and implementing important solutions to protect sensitive data from malicious persons. Thereby various measures of protection have been developed and aim to minimize the risks and damages of attacks. Among them, cryptography plays a vital and crucial role in protecting sensitive transmissions and electronic exchanges through complex networks. Numerous scientific studies have emerged with the advent of the cloud and the Internet of Things (IoT); all of them have expressed a strong need for building secure, efficient and fast cryptosystems targeting confidentiality, integrity and authentication. The last two objectives are essentially built on hash functions which are the main components of many applications and secure networks. The purpose of this paper is to give recent advances of lightweight cryptographic solutions that meet the requirements of constrained systems, and to present a study, in terms of security, energy-consuming and efficiency, of the main hash functions standardized by NIST (National Institute of Standards and Technology). In the end, the paper will give a comparison between the studied hash functions aiming to come up with a recommendation of good lightweight hash functions suitable for implementation in an IoT framework.

Keywords: Constrained Environments; Hash function; IoT; Lightweight Cryptography.
2010 MSC: MSC code1, MSC code2, more.

1. Introduction

The impressive technological revolution and its rapid growth have greatly changed our way of life; we are living in a connected world in which we handle daily binary data transiting through a complex ecosystem, smart devices communicating through a multitude of networks, etc. Most of the internet activities concern the exchange of different types of data; some of them are critical such as login, passwords, credit card numbers, etc. Exchanged data are facing different hazards; they can be intercepted, deleted, changed/encrypted, or sold for money, etc. Thus, almost all sectors such as transport, energy, e-commerce, hospitals, health industry, and education or government institutions are vulnerable to cybercrime; the injection of viruses, the spread of malware constitutes a threat that can damage or paralyze vital sectors. Otherwise, we can note that hackers techniques constantly evolve with technology, and types of attacks are constantly mutating and adapting to defense mechanisms. Therefore, strengthening and innovating security mechanisms is of great importance for the survival of the digital world.

*Corresponding author
Email address: slazaar@uae.ac.ma (Saiida Lazaar)

doi:10.31559/glm2021.10.2.4

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

To avoid or to minimize the risks of attacks, various solutions of protection have been developed as for instance access policy, firewalls, anti-viruses, anti-malwares, IDS (Intrusion Detection Systems). These solutions are not sufficient because some existing systems, applications, and protocols present often weaknesses likely to be exploited by cybercriminals. Cryptography plays a vital and crucial role for protecting sensitive transmissions and reinforcing existing solutions [1]. Furthermore, cryptography owes its strength from important mathematical areas such as arithmetic, number theory, algebra with Galois Fields, etc. [2]. This force is often based on mathematical conjectures difficult to be demonstrated, namely discrete logarithm problem, or factorization of very large integers into a product of prime numbers. Since the past decades, scientists community expresses a strong need for building cryptosystems satisfying: efficiency, speed, security, and resistance against cryptanalysis coming essentially from active attacks; the need has greatly increased with the advent of IoT [13]. These cryptosystems are often targeting confidentiality, authentication; and integrity [3]. Integrity ensures that data are not corrupted during their transmission and it is essentially built on hash functions which are the main components of many security applications and communication protocols on Ipv4/Ipv6 [4, 5] like SSL (Secure Sockets Layer), TLS (Transport Layer Security) and IPSec (IP Security). The present work aims to present an analysis of lightweight block ciphers and lightweight hash functions suitable for implementation on resource-limited computational devices. A focus will be done on some selected hash functions qualified as lightweight, essential for building authentication and integrity solutions needed for securing constrained environments.

The rest of the paper is organized as follows: Section 2 presents an overview on lightweight cryptography including a discussion on the most important lightweight block ciphers essentially those standardized by the NIST; a discussion on security aspects is given. Section 3 is devoted to a review on hash functions, including conventional and cellular based hash functions. Section 4 is dedicated to the description of our proposed solution. We end this paper with a conclusion and some outlines.

2. Lightweight Cryptography

To ensure the confidentiality and integrity of data, a list of cryptosystems has been released in modern cryptography. The most important ones are of two categories. The first category concerns symmetric algorithms using the same secret key for encrypting and decrypting data as for instance DES (Data Encryption Standard), AES (Advanced Encryption Standard), 3DES etc. [3]. Their role is to encrypt and to decrypt data into blocks through a number of iterations/rounds; these iterations use transformations versus some sub keys generated from the chosen single secret key. The second category is asymmetric algorithms using public keys to cipher texts and secret keys to reconstruct plaintexts; the well-known algorithms are RSA, El Gamal, and elliptic curve-based ciphers, [3].

With the advent of IoT and smart citie applications, a great need is to secure small and smart devices, but conventional cryptography standards know some limits making them difficult to be implemented on constrained devices including embedded systems deployed in various industrial installations, smart cards, RFID and sensor networks. Another type of cryptography has emerged within a new subfield of cryptography named lightweight cryptography setting the challenge to build secure solutions on hardware and software, and tailored for each constrained device. The properties of lightweight cryptography are presented in ISO/IEC JTC 1/SC 27 and in ISO/IEC 29192. On software, smaller code and RAM size are recommended for lightweight applications in ISO/IEC 29192, while chip size and energy consumption constitutes important measures in lightweight solutions on hardware. More details can be found in NIST report [6] which presents a study demonstrating the performance of lightweight block ciphers over conventional block ciphers taking into account, smaller block sizes, smaller key sizes, simpler rounds, simpler key schedules, and minimal implementations of only necessary functions for consuming minimal resources.

2.1. Lightweight block ciphers and security analysis

This section discusses the performance of some selected lightweight block ciphers: AES-128, DESL, PRESENT, SIMON, SPECK, RC5, TEA and XTEA; it may be possible to apply them for protecting data on

constrained environments as we will see here after.

A number of lightweight block ciphers have been proposed targeting good performance as for instance AES, standardized by NIST and more precisely AES-128. The DES algorithm has also been adapted for lightweight cryptography applications. For example, DESL (DES lightweight extension) is a variant of DES where initial and final permutations are excluded, and where a single S-box is used instead of eight; the omission of permutations is for improving the size of the hardware implementation. Concerning DESL/DES security, a brute force attack is able to break the entire 56-bit key. Linear, differential and hybrid differential-linear cryptanalysis applied on DES/DESL can led to breaking the 16 rounds of the algorithm. PRESENT is one of the first lightweight block cipher built for constrained hardware environments. The 4-bit S-box used in PRESENT requires 28 GEs (Gate Equivalence) whereas the AES S-box required 395 GEs. For this reason, PRESENT is about 2.5 times smaller than AES: Thanks to this performance, the International Organization for Standardization and the International Electrotechnical Commission introduced PRESENT in the new international standard for lightweight cryptographic methods. Concerning the security, PRESENT is threatened by a differential cryptanalysis. SIMON and SPECK are families of lightweight block ciphers designed by the NSA (National Security Agency) to be simple, flexible, and perform well in hardware for SIMON, and in software for SPECK. The two ciphers are expected to operate well on various IoT devices. We note that Speck is vulnerable to differential and side-channel attacks, and SIMON is vulnerable to differential and linear Hull cryptanalysis. Otherwise, some old algorithms like RC5 (symmetric-key block cipher), TEA (Tiny Encryption Algorithm) and XTEA (eXtended TEA) are suitable for constrained software environments because they are built on simple round structures [6].

3. Hash functions

To begin this section, it would be preferable to remind the concept of hash function.

Definition 3.1. Let Σ^* be a set of alphabetic strings. We define a hash function h as an application from Σ^* to $\Sigma^n, n \in \mathbb{N}$ such that it associates images strings of a fixed length to strings of any length in Σ^*. The function h cannot be injective.

Definition 3.2. The function h is called a one-way hash function if it satisfies the following conditions:

- The argument x can be of arbitrary length and the image $h(x)$ has a fixed length of n bits.
- For a given y image of the hash function, it is almost impossible to find a message x such that $h(x) = y$. Given x and $h(x)$, it is very difficult or impossible to find an arguement $x^{'} \neq x$ such that $h(x^{'}) = h(x)$.

3.1. Brief review on conventional hash functions

A cryptographic hash function can be applied to verify data integrity, message authentication and digital signatures; it corresponds to a one-way function where the input is an arbitrary block of data and where the output is of fixed size. The encoded data is called the message, and the hash value is the message digest [8]. The most known cryptographic hash functions are MD5 (Message Digest 5), SHA-1 (Secure Hash Algorithm 1), SHA-256, SHA-512, RIPEMD-160, and HAVAL; all these functions are based on the so-called MD4, [9].

In 1991, Ron Rivest developed MD5, a 128-bit hash function more secure than the initial hash function MD4. MD5 knew some difficulties in 1993 related to its compression operation; it suffered vulnerabilities due to collision attacks; the attacks were also applied to several other hash functions built on MD4 like HAVAL and RIPEMD [10].

SHA-1 successor of MD4 was designed by the NSA and produces a 160-bit hash value. Due to its vulnerabilities against collision attacks, many organizations have recommended to replace SHA-1 by SHA-256.

Several cryptographic hash algorithms were attacked in 2004-2005; the attacks revealed the weaknesses of existing hash functions and led in 2007 to a new hash function adopted by NIST named SHA-3. The SHA-3 family corresponds to the following cryptographic hash functions, named SHA3-224, SHA3-256, SHA3-384, and SHA3-512, SHAKE-128 and SHAKE-256. SHA-3 functions are based on the Keccak sponge function, designed by G. Bertoni, J. Daemen, M. Peeters, G. Van Assche [24]. Keccak was the winner of the NIST hash function Competition held on 2012 [11]. In 2013, two sets of lightweight implementations of all SHA-3 were presented. The final selection, implemented on Xilinx devices are BLAKE-256 followed by Grostl [12].

3.2. Cellular automata-based hash functions

Cellular automata (CA) were first invented by Ulam and Von Neumann in 1940; afterwards, the CA theory was developed by Stephen Wolfram author of many reference studies in the field of CA [26]. CA is a dynamical system of a network of cells evolving versus given rules and according to specific neighborhoods. CA can be applied in many fields as biology, transport, biomathematics, cryptography, network security, etc. For example, CA are a useful tool to design IDS (Internet Detection Systems) and can led to interesting schemes for malware propagation modelling [26]. In Cryptography, CA were applied to design secure and fast cryptosystems to guarantee confidentiality, integrity or authentication in electronic transmissions [15, 28, 29].

Definition 3.3. A cellular automaton (CA) is a dynamic system considered as a discret model equal to a regular grid of cells defined by its dimension, a set of finite states, a neighborhood and a set of rules. If A is a cellular automaton, then A can be expressed by the set $\{S, Z^d, V, f\}$ where S is a finite set of states, d is the size of the automaton, Z^d is the space of the automaton, and f is the rule also called transition function defined from S^n to S with $n = card(V)$, and V is the set of neighborhood.

Cryptographic hash functions based on CA are suitable for applications where data must to be authenticated and where CPU time of transmission can be very important as for instance applications related to secure transmissions between smart devices on IoT. Damgard in [23] built a fast and collision free one way hash function based on CA. And in [16], authors presented a study demonstrating that based on CA, a new one-way hash function providing authentication and data integrity is suitable for fast implementation in hardware; and it is secure against all known attacks. The characteristics of this function make it useful for securing smart cards and electronic cash payment protocols. In 2013, a detailed study demonstrated that CA based schemes are able to define hash functions for low hardware complexity on small silicon area [10]. In 2014, a new CA hash function called CASH was presented and a comparison with some CA based hash functions has been given and demonstrated that the proposed hash function preserved all good characteristics of the previous CA based hash function guarantying improvement of security and complexity, The results show that CASH is better than SHA-3 with respect to throughput [14]. A more recent study published in 2017 presented a fast new CA based hash function compared to the well-known hash functions SHA2-512, MD5 and keccak. Numerical tests were performed on simple configuration machine (Intel Core i5, 64-bit, 4 GB, 1.8Ghz). The results meet the integrity and authentication requirements and show that the new hash function is resisting against forgery attacks [17]. Following the previous scientific contributions, we can conclude that CA based hash function can perform well than MD5, SHA-512 and SHA-3 when implemented for specific requirements. CA can lead to interesting solutions of authentication and integrity on constrained environments.

3.3. Analysis and comparison of the well-known lightweight hash functions

The expected usage of conventional and lightweight hash functions differs in various aspects such as smaller internal state, output sizes and smaller message size. Conventional hash functions may not be suitable for constrained environments, mainly due to their large internal state sizes and high-power consumption requirements. The most interesting lightweight hash functions are: PHOTON, SPONGENT,

Lesamnta-LW, Quark, Keccak, Gluont, Neiva, Armadillo, BLAKE, SHA-1, SPECK, SHA-3; MD5. An analysis of these functions is given here after. Firstly, following a study on [18], we can notice that PHOTON is recommended for constrained devices like passive RFID tags. Otherwise, SPONGENT [19] is a family of lightweight hash functions with hash sizes of 88,128, 160, 224, and 256 bits based on a sponge construction. The importance of these functions lies in the fact that it leads to fast algorithms.

Hirose et al. defined a lightweight 256-bit hash function under the name of Lesamnta-LW respecting security against collision, preimage, and second preimage attacks and using AES as the compression function [20].

In [22], authors presented the hash function family Quark based on the sponge construction. It was demonstrated that Quark is resisting against well-known attacks, and gave satisfactory performance when implemented on hardware with minimal memory requirements. In 2017 [21], a classification of lightweight cryptographic hash algorithms has been presented and aimed to select a lightweight cryptographic primitive ensuring security and respecting resource constraints. A performance comparison of Photon, Quark, Keccak, Gluont, Spongent, Neiva, Armadillo and Lesmanta has been presented considering specific metrics such as throughput, power, and, hardware efficiency. This study did not ended to a clear classification between the analyzed functions. However, this study could be useful in making a compromise between security and environment constraints. More recently, in 2019, authors presented software implementation benchmarks for various hash functions: MD5, BLAKE, SHA-1, SPECK, and SHA-3. The implementation was performed for resource limited devices [27]. The benchmarks were evaluated on a batteryless RFID device and revealed that MD5 produces the best performance when compared to BLAKE, SHA-1, SPECK, and SHA-3.

4. Definition of a lightweight cryptography based solution to secure resource-constrained environment communications

In this section, we define a global lightweight solution protecting resource-constrained environments, targeting confidentiality, integrity and authentication. This solution includes:

- A confidentiality mechanism to protect exchanged data using AES-128 associated to a secret key K of 128-bits (a lightweight block cipher implemented with block sizes of 128-bits);
- A secure mechanism to share the secret key K using the algorithm RSA-1024 which is considered as a lightweight algorithm;
- A lightweight hash function as for instance Quark function to enhance the security process guaranteeing integrity and authentication.

Let consider two communicating entities A as a sender and B as a receiver. Firstly, using for example MQTT (Message Queuing Telemetry Transport) protocol, B sends to A his couple of public keys $(e, n) \in \mathbb{N} \times \mathbb{N}$; the private key $d \in \mathbb{N}$ is kept on B server. The sender A encrypts the plaintext text M using AES-128 with the key K producing the cipher text C. After this step, the system call RSA algorithm to encrypt the key K using the equation 4.1:

$$K' = K^e \mod n \tag{4.1}$$

Thereafter, the lightweight function Quark is called to hash C and to produce $hash(C)$. Following this step, the entity A sends to B the block: $(C, hash_A(C), K')$.

From the other part, the receiver executes the following steps:

- Deduction of the key K from K' using RSA with the equation :

$$K = K'^d \mod n \tag{4.2}$$

where d is the private key of B.

- Decryption of C using the algorithm AES and deduction of a plaintext text M_1.
- Call of Quark function to hash C and to produce $hash_B(C)$.
- Comparison of $hash_B(C)$ and $hash_A(C)$:

If $hash_B(C) = hash_A(C)$, then accept the data and confirm that $M_1 = M$; and if $hash_B(C) \neq hash_A(C)$, then reject the data.

Note that the proposed solution is a global one as it offers data protection, verification of integrity and authentication.

To summarise, the proposed solution is illustrated in figure 1.

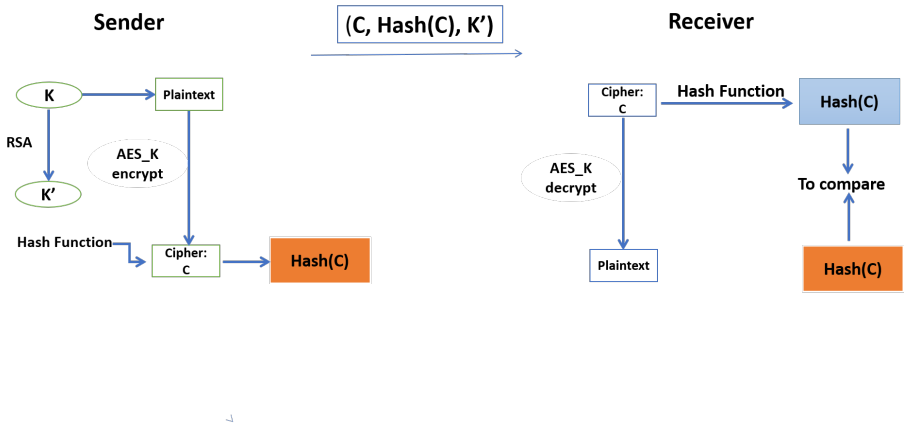

Figure 1: A lightweight cryptography based solution to secure transmissions on resource-constrained environments.

5. Conclusion and some directives for a future work

In this work, we presented and discussed the most relevant lightweight block ciphers and lightweight hash functions including those based on cellular automata. The work covered performance analysis of the most known block ciphers. A particular focus was accorded to the main hash functions, qualified as lightweight, and suitable for building authentication and integrity solutions essential to secure constrained environments. As we noticed above, MD5 produced the best performance when compared to BLAKE, SHA-1, SHA-3, and SPECK. Otherwise, Quark can be considered as a good candidate for integrity and authentication solutions since it is resisting against well-known attacks, and presents good performance when implemented on hardware with minimal memory requirements. Regarding CA applications, we can affirm that CA based hash function can perform well than MD5, SHA-512 and SHA-3 when implemented for specific requirements. In a future work, we will carry out more simulations on the proposed secure solution that could represent a model implemented on a network including different layers as smart devices, gateway, and servers on the cloud. The model can be applied for many applications as for instance the protection of sensitive data exchanged through a health architecture.

Acknowledgments

The author gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] Tom St Denis, Simon Johnson. Cryptography for developers. Syngress Publishing, Inc 2007 1
[2] Leonard Eugene Dickson. Linear Groups With an Exposition of the Galois Field Theory. Dover Phoenix Editions, 2003 1
[3] Gilles Brassard. Modern cryptology. Springer-Verlag, 1988 1, 2
[4] Silvia Hagen. IPv6 Essentials. OReilly, 2006 1

[5] Drago Zagar, Kresimir Grgid, Snjezana, Rimac-Drlje. Security aspects in IPv6 networs-Implementation and testing. Computers and Electrical Engineering, Volume 33, Issues 5-6, 2007 1
[6] NIST REPORT 8114 https://doi.org/10.6028/NIST.IR.8114 2, 2.1
[7] Muzaffar Rao, Thomas Newe and Ian Grout. Secure Hash Algorithm-3 (SHA-3) implementation on Xilinx FPGAs, Suitable for IoT Applications. Proceedings of the 8th International Conference on Sensing Technology, Sep. 2-4, 2014, Liverpool, UK
[8] Jongsung Kim, Alex Biryukov, Bart Preneel, Seokhie Hong. On the Security of HMAC and NMAC Based on HAVAL, MD4, MD5, SHA-0 and SHA-1. International Conference on Security and Cryptography for Networks, book series (LNCS, volume 4116): pp 242-256. ISBN 978-3-540-38081-8 3.1
https://doi.org/10.1007/11832072$_1$7
[9] Jun-Cheol Jeon. Analysis of Hash Functions and Cellular Automata Based Schemes. International Journal of Security and Its Applications. Vol. 7, No. 3, May, 2013 3.1
[10] Morris J. Dworkin. SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions. Federal Inf. Process. Stds. (NIST FIPS) 2015 3.1, 3.2
[11] Kaps JP. et al. (2011) Lightweight Implementations of SHA-3 Candidates on FPGAs. In: Bernstein D.J., Chatterjee S. (eds) Progress in Cryptology INDOCRYPT 2011. INDOCRYPT 2011. Lecture Notes in Computer Science, vol 7107. Springer, Berlin, Heidelberg. 3.1
[12] Medaglia C.M., Serbanati A. An Overview of Privacy and Security Issues in the Internet of Things. In: Giusto D., Iera A., Morabito G., Atzori L. (eds) The Internet of Things. Springer, New York, NY, 2010 3.1
https://doi.org/10.1007/978-1-4419-1674-7$_3$8
[13] Kuila S., Saha D., Pal M., Chowdhury D.R. (2014) CASH: Cellular Automata Based Parameterized Hash. In: Chakraborty R.S., Matyas V., Schaumont P. (eds) Security, Privacy, and Applied Cryptography Engineering. SPACE 2014. Lecture Notes in Computer Science, vol 8804. Springer, Cham 1
[14] Said BOUCHKAREN and Saiida LAZAAR. A New Iterative Secret Key Cryptosystem Based on Reversible and Irreversible Cellular Automata. International Journal of Network Security, 18.2 (2016): 345-353. 3.2
[15] Mihaljevic M., Zheng Y., Imai H. (1998) A cellular automaton based fast one-way hash function suitable for hardware implementation. In: Imai H., Zheng Y. (eds) Public Key Cryptography. PKC 1998. Lecture Notes in Computer Science, vol 1431. Springer, Berlin, Heidelberg 3.2
https://doi.org/10.1007/bfb0054027
[16] Bouchra Echandouri et al.. Keyed- CAHASH: a New Fast Keyed Hash Function based on Cellular Automata for Authentication. International Journal of Computer Science and Applications Technomathematics Research Foundation. Vol. 14, No. 2, pp. 164 - 180, 2017 3.2
[17] Guo J., Peyrin T., Poschmann A. (2011) The PHOTON Family of Lightweight Hash Functions. In: Rogaway P. (eds) Advances in Cryptology CRYPTO 2011. CRYPTO 2011. Lecture Notes in Computer Science, vol 6841. Springer, Berlin, Heidelberg. Online ISBN 978-3-642-22792-9 3.2
https://doi.org/10.1007/978-3-642-22792-9$_1$3
[18] Bogdanov A., Knezevic M., Leander G., Toz D., Varici K., Verbauwhede I. (2011) spongent: A Lightweight Hash Function. 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. Online ISBN 978-3-642-23951-9 3.3
[19] S. Hirose, K. Ideguchi, H. Kuwakado, T. Owada, B. Preneel, and H. Yoshida. A lightweight 256-bit hash function for hardware and low-end devices: lesamntalw. In International Conference on Information Security and Cryptology. Springer, 2010, pp. 151168. 3.3
[20] Baraa Tareq Hammad et al. A survey of Lightweight Cryptographic Hash Function. International Journal of Scientific & Engineering Research Volume 8, Issue 7, July-2017 806 ISSN 2229-5518 3.3
[21] Aumasson, JP., Henzen, L., Meier, W. et M. Naya-Plasencia. Quark: A Lightweight Hash. Journal of Cryptology (2013) Volume 26: pp 313-339. https://doi.org/10.1007/s00145-012-9125-6 3.3
https://doi.org/10.1007/s00145-012-9125-6
[22] I. B. Damgard. A design principle for hash functions, Springer-Verlag Conference on CRYPTO, USA, pp. 416-427, 1989 3.3
[23] Bertoni G., Daemen J., Peeters M., Van Assche G. (2013) Keccak. In: Johansson T., Nguyen P.Q. (eds) Advances in Cryptology EUROCRYPT 2013. EUROCRYPT 2013. Lecture Notes in Computer Science, vol 7881. Springer, Berlin, Heidelberg 3.2
[24] Al-Hamami, Alaa Hussein. Handbook of Research on Threat Detection and Countermeasures in Network Security. IGI Global, 31 oct. 2014 - 450 pages. 3.1
[25] Stephen Wolfram. Cellular Automata And Complexity. CRC Press 8 March 2018
[26] Yang Su, Yansong Gao, Omid Kavehei, Damith C. Ranasinghe. Hash Functions and Benchmarks for Resource Constrained Passive Devices: A Preliminary Study. 2019 IEEE Percom Workshops. https://arxiv.org/pdf/1902.03040.pdf 3.2
[27] Said Bouchkaren, Saiida Lazaar. CAES Cryptosystem: Advanced Security Testsand Results International Journal of Network Security, Vol.20, No.1, PP.177-183, Jan. 2018 (DOI: 10.6633/IJNS.201801.20(1).19)177 3.3
[28] Sbaytri Y., Lazaar S., Benaboud H., Bouchkaren S. (2019) A New Secure Cellular Automata Cryptosystem for

Embedded Devices. In: Renault E., Boumerdassi S., Leghris C., Bouzefrane S. (eds) Mobile, Secure, and Programmable Networking. MSPN 2019. Lecture Notes in Computer Science, vol 11557. Springer, Cham 3.2

[29] Adrian Hernandez-Becerril, Mariko Nakano-Miyatake, Hector Perez-Meana. A Parallel Authenticated Encryption Sharing Scheme Based on Cellular Automata. Proceedings of the World Congress on Engineering and Computer Science 2014 Vol I WCECS 2014, 22-24 October, 2014, San Francisco, USA 3.2