Lightweight Cryptography: an IoT Perspective

Deena Nath Gupta, Rajendra Kumar

Abstract: ***We need a secure environment in order to communicate without any information leakage. From large devices having UPS to small devices having a battery, the parameter about security changes over time. We need to work in three basics of security: (1) Mutual authentication between devices, (2) Strong encryption methodology for transmission, and (3) Secure storage environment with anytime availability.***

The IoT enabled devices demands a lightweight secure environment. In this paper, we are concerning only about the second point, i.e. Strong encryption methodology for transmission. We will study some of the methods related to lightweight cryptography; will talk about different issues in the field of secure transmission; and will try to find out some research gap with a possible countermeasure.

Index Terms**:** ***IoT, Lightweight Cryptography, Hash Function, Random Number Generator.***

I. INTRODUCTION

Internet of Things can be seen as the combo of these three visions: Things-oriented, Internet-oriented, and Semantic-oriented [1]. A statement is issued by CASAGRAS (Coordination and Support Action for Global RFID-related-Activities and Standardization) consortium – "A global infrastructure to connect physical and virtual objects is known as the Internet of Things." Wireless Identification and Sensing Platform (WISP) have been used to measure quantities (like light, temperature, acceleration, strain, and liquid level) in a certain environment. The workflow of the Internet of Things can be defined as: Object sensing, identification and communication of object-specific information then trigger an action, and at last results of action invoked. Chonggang wang and Mohamoud Daneshmand suggested the Internet of Things as a cyber-physical system [2]. A cyber-physical system can be seen as a network of networks where the collection of raw data should be done with utmost care.

Many architectures of the Internet of Things have been suggested and from them, Cisco's seven-level model is the most famous [3]. Earlier we were having a three-level model that consists of the wireless sensor network, cloud servers, and applications. After that, there comes a five-level model which includes edge nodes, object abstraction, service management, service composition, and applications. The currently used Cisco's seven-level model consists of edge nodes, communication, edge computing, data accumulation, data abstraction, applications, and users and centers. Mainly used devices to identify and/or collect the information from objects are RFID, 2D-barcode, and infrared sensors, IEEE 802.15.4. The transmission technologies used in the Internet of Things arena are 3G, UMTS, Wi-Fi, Bluetooth, infrared, and Zigbee etc. The service-oriented architecture consists of the application, service composition, service management, object abstraction, and trust, privacy and security management [4].

The rising popularity and acceptance of the Internet of Things can be realized as now we are moving towards (a) Transportation and Logistic Domain (with the applications in (i) Logistics, (ii) Assisted Driving, (iii) Mobile Ticketing, (iv) Environmental parameters Monitoring, and (v) Augmented Maps), (b) Healthcare Domain (with the application in (i) Tracking, (ii) Identification and Authentication, (iii) Data Collection, and (iv) Sensing), (c) Smarts Environments Domain (with the application in (i) Comfortable homes and offices, (ii) Industrial plants, and (iii) Smart Museum and gym), (d) Personal and Social Domain (with the applications in (i) Social Networking, (ii) Historical Queries, (iii) Losses, and (iv) Thefts), (e) Futuristic Applications Domain (with the applications in (i) Robot Taxi, (ii) City Information Model, (iii) Enhanced Game Room), (f) Application in Agriculture, (g) Water Scarcity Monitoring, (h) Energy Management, (i) Construction Management, and many more [5] [6] [7].

IoT is becoming a universal dependable technology. It is given greater responsibilities and to be a responsible technology it should work on odds coming in its way. In terms of the Internet of Things we still need to work on (a) Standards, (b) Mobility Support, (c) Naming and Identity Management (Assigning an IPv6 address to each element), (d) Object safety, (e) Transport Protocol, (f) Traffic Characterization and QoS Support, (g) Authentication, (h) Data Integrity, (i) Information Privacy, and (j) greening of IoT [8] [9].

Some agencies which are working as the key development force for IoT are Microsoft's Eye-on-Earth platform, Cluster of European Research Project on the IoT, The Internet of Thing Architecture (IoTA), IoT@work, IoT-initiative (IoT-i), European Research Cluster on the IoT, and many more [10]. These agencies are working continuously to convert the cryptographic algorithms into their lightweight version.

The rest of the paper is organized as follows: Section 2 will talk about the lightweight cryptography and also show a comparison between some well-known ciphers, in section 3 we will talk about the algorithms comprises of some features that tend the existing cryptographic algorithms into their lightweight variant. We are mainly considering Hash functions and Random numbers for this purpose. Some results and discussion are there in Section 4. Section 5 will provide the conclusion along with some future research direction.

Revised Manuscript Received on June 07, 2019.

Deena Nath Gupta, Research Scholar, Department of Computer Science, Faculty of Natural Sciences, Jamia Milla Islamia, New Delhi-110025, India.

Rajendra Kumar, Associate Proferssor, Department of Computer Science, Faculty of Natural Sciences, Jamia Milla Islamia, New Delhi-110025, India.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

II. LIGHTWEIGHT CRYPTOGRAPHY

Lightweight cryptography is a security system that is made for constrained devices. While making any lightweight algorithm, the main focus is on its hardware implementation. The required logic gate to run any program is termed as Gate Equivalent. The lower the GE, the lighter the algorithm is. In TABLE 1, a comparison of some well-known ciphers is given in terms of Gate Equivalent. On the other hand, while working on codes we try to make them as small as we can without compromising the security. The software in this mechanism should be compatible with the tiny OS used in small battery operated devices. As we progress in automation and start including the devices from our daily life, the concern for their security also risen [11].

To provide adequate security to these devices the concept of lightweight cryptography emerged. Some advancement from our traditional block and stream ciphers includes the concept of shift operations. In the process of lightweight security system design, we mainly use the concept of low computing overhead, pre-image resistance and second pre-image resistance hash functions and pseudorandom number generators that include the concept of non-linear feedback shift registers. Lightweight cryptography is important to save the energy and storage of devices. It is very useful for low-power embedded systems, machine to machine communication, radio frequency identification tags, nanotechnology, sensors, and smart networks [12] [13].

Block Cipher	Key Size (in bits)	Block Size (in bits)	Gate Equivalent
PRESENT-80	80	64	1570
PRESENT-128	128	64	1886
AES	128	128	3400
HIGHT	128	64	3408
mCrypton	96	64	2681
DES	56	64	2309
DESL	56	64	1848
DESXL	184	64	2168
Hummingbird	128	16	2159
Trivium	80	1	2580
Trivium X 8	80	8	2952
Trivium X 16	80	16	3166
Grain	80	1	1450
Grain X 8	80	8	2756
Grain X 16	80	16	4248
MICKEY	128	1	5039
Pandaka (16,6)	96	16	760
Pandaka (32,6)	192	32	1520

Table 1: A comparison of some well-known ciphers is given in terms of Gate Equivalent.

III. ALGORITHM FOR LIGHTWEIGHT CRYPTOGRAPHY

A. Hash Function

Hash functions are mainly used for authentication, for example, message authentication code. Because of this only feature, Hash functions are widely used in cryptography. The development of Hash calculation is no way different from the development of secure key calculations. As the computer industry progressed, researchers tried for lightweight hash design. As these designs affect the hardware implementation, the community decides some criteria for lightweight hashing, i.e., if the algorithm takes nearly 2000 GE (Gate Equivalent) then only we will term it as lightweight [14]. A comparison of earlier hash designs having Gate Equivalent above 5,000 is given in TABLE 2.

To achieve lower Gate Equivalent, a trade-off between creating new schemes and reusing available schemes (according to power constrained system) is much needed. A hash function is mainly determined by the number of state bits and the size of functional and control logic used in a ROUND function. To achieve the low size and low power constraints of tinny devices the focus should be on state size because logic size does not dominate the total area requirements of the design much.

Digest	Hash	Gate Equivalent	ASIC
512-bit digest	**Cube Hash (SHA-3)**	**7630 GE**	**0.13 μm**
128-bit digest	**Feldhofer & Wolkerstorfer (MD-5)**	**8001 GE**	**0.35 μm**
160-bit digest	**O'Neill (SHA-1)**	**6122 GE**	**0.18 μm**
256-bit digest	**Yoshida et. al. (MAME Composition)**	**8100 GE**	**0.18 μm**

Table 2: A comparison of earlier hash designs having Gate Equivalent above 5,000

Unlike pervasive computing, tinny devices can work on 64 or 80-bit security. Sponge function is a new way of building hash functions in which the internal state S of t bits consists of C-bit capacity and the r-bit bitrate ($t = c + r$), is first initialized with some fixed value. Sponge function can also be used as a message authentication code. Hence, we use sponge function to minimize the number of memory registers requirement in hardware. A comparison of Lightweight Hash Designs that are using Sponge Construction is given in TABLE 3.

Some of the already designed lightweight hash functions are QUARK, PHOTON, SPONGENT, GLUON, AND Hash-One [15]. QUARK's design methodology was based on Grain (a stream cipher) and KATAN (a block cipher). Three instances of QUARK exists, those are U-QUARK, S_QUARK, and D-QUARK [16]. Next is SPONGENT, which is based on PRESENT-type permutation. Unlike QUARK, SPONGENT produces fixed length output [17]. Many variants of SPONGENT are present and they can be referred to as SPONGENT-n/c/r. Where n denotes the hash size, c denotes the capacity, and r denotes the rate.

PHOTON is also a sponge-based construction in which a matrix with 8-bit entries is used to represent the internal state. PHOTON uses AES like fixed key permutation [18]. 12 rounds of Add Constants, Sub Cells, Shift Rows, and Mix Column Serial are used just like in AES.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Digest	Hash	GE (Gate Equivalent)	ASIC (Application -Specific Integrated Circuit)
64-bit digest	Bogdanov et. al. DM-Present 2008	1600 GE (LW)	0.18 µm
64-bit digest	KECCAK 2010	2520 GE (LW)	0.13 µm
64/ 80/ 112 bit digest	U/ D/ S QUARK (CHES-2010)	1379/ 1702/ 2296 GE (LW)	0.18 µm
80/ 128/ 160/ 224/ 256 bit digest	PHOTON (CRYPTO -2011)	865/ 1122/ 1396/ 1736/ 2177 GE (LW)	0.18 µm
80-bit digest	SPONGENT (CHES-2011)	1329 GE (LW)	0.13 µm
80-bit digest	GLUON	2799 GE (LW)	0.18 µm
80-bit digest	Hash-One	1006 GE (LW)	0.18 µm

Table 3: A comparison of Lightweight Hash Designs that are using Sponge Construction

GLUON is inspired by F-FCSR-v3 and X-FCSR-v2 (both are stream ciphers) [19]. A word ring FCSR that include the main shift register and a carry register is used to design GLUON. Two NFSRs of sizes 80 bits and 81 bits are used in Hash-One. Hash-One uses the sponge state of 161-bits. 324 rounds of state updates needed in the absorption of first and last message bits whereas only 162 required for intermediate message bits. Squeezing phase need only one round of state updates.

Hash-One uses shift registers to reduce complexity. The hash function should be tested for statistical randomness, collision resistance, strict avalanche criteria (SAC), linear span test and coverage test. SAC states that for a particular S-box, whenever one input bit is changed, every output bit must change with probability 0.5. Collision resistance means that it should be hard to find two messages with the same hash value. The coverage test evaluates a given function f through examining the size of the output set formed from a subset of its domain. A comparison of Lightweight Hash Designs on the basis of the number of cycles is given in TABLE 4.

It is widely accepted that Hash Functions are used to build a highly secure system. Two of its features, namely pre-mage resistance and second pre-image resistance hash design, make this function unbreakable. As much as the vulnerability is concerned, everyone relies on the functionality of hash designs. A lightweight version of hash was much required as we moved towards the era of a lightweight to ultra-lightweight cryptography. To work on constrained devices, a new method to calculate the hash is generated. The new method of performing hash is Sponge. The process of sponge construction is depicted in FIGURE 1. The sponge used to build an input and output function of variable length using fixed length permutation "f" that operates on "b" number of bits. Further "b" (the width) can be divided in bit rate and capacity. "b = r + c" bits. Two non-linear feedback shift registers and a linear feedback shift register are used to construct the sponge construction which is updated using three different non-linear functions [20].

There are two phases: the first is absorbing and second is squeezing [21]. In the first phase, "r bits" of the state and the "r bit" input message blocks are XORed. In the second phase, the first "r bits" of the state are returned as output blocks. We can design a permutation "f" on "b = r + c bits" to build the sponge function F(f, pad,r) with domain Z_2* and codomain Z_2^{∞} .

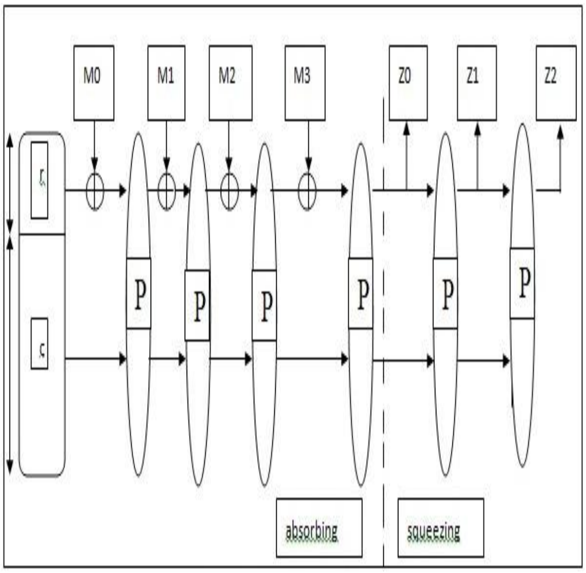

Fig 1: Sponge Construction

The input and output size is arbitrarily long in a sponge and hence this construction is used for a hash function, stream cipher or a MAC design. The output of a sponge is taken as the first l bits which requested.

For Z = sponge [f, pad, r](m, ℓ), we have:

P = M || pad[r] (|M|)

S = ABSORB [f, r] (P)

Z = SQUEEZE [f, r] (s, ℓ)

The absorbing function ABSORB (f, r) takes a string P as input with |P| multiple of r. the output of this function is the state obtain after absorbing P. From the state,

Hash function	n	C	R	Preimage	Collision	Second Preimage	Process (µm)	Area (GE)	Cycles
Hash-One	160	160	1	160	80	80	0.18	1006	324/162

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Hash-One	**160**	**160**	**1**	**160**	**80**	**80**	**0.18**	**2130**	**14/7**
SPONGENT	**176**	**160**	**16**	**144**	**80**	**80**	**0.13**	**1329**	**3960**
SPONGENT	**176**	**160**	**16**	**144**	**80**	**80**	**0.13**	**2190**	**90**
D-QUARK	**176**	**160**	**16**	**160**	**80**	**80**	**0.18**	**1702**	**704**
D-QUARK	**176**	**160**	**16**	**160**	**80**	**80**	**0.18**	**2819**	**88**
PHOTON	**160**	**160**	**36**	**124**	**80**	**80**	**0.18**	**1396**	**1332**
PHOTON	**160**	**160**	**36**	**124**	**80**	**80**	**0.18**	**2117**	**180**
GLUON	**160**	**160**	**16**	**160**	**80**	**80**	**0.18**	**2799**	**50**

Table 4: Comparison of Lightweight Hash Designs on the basis of the number of cycles

in starting we truncate ℓ bits; this is done via squeezing phase. We can't get back P from s or s from Z, this is the beauty of sponge constructions, and it preserves backward secrecy.

B. Random Number Generator

Random number generators are of much importance because of their uses in confidential key generation. A type of challenge-response methodology may also be created by using random numbers. They are also helpful in the nonce-based authentication system. By using a nonce, we can secure our devices from any type of hardware attack. Small battery operated devices are very much in need of the algorithms that can work on shift operations and can produce a desirable level of security. To some extent, one can use a hash function and other encryption methods but pseudo-random number generator is a major security component for small battery-operated devices [22].

To generate truly random numbers based on a physical source (some well-known methods are: thermal noise ZENER diode and radioactive decay).

However, they are inefficient in terms of aggregating many physical resources. On the other hand, pseudo-random number generators can be generated mathematically by using some functions such as Linear Congruential Generators (LCG) and Linear Feedback Shift Registers (LFSR). In cryptography, we use LFSRs very often to perform shift operations because of its efficiency and simplicity in terms of implementation. To check this distribution, we perform a test that focuses on the existing non-randomness in the generated PRNG binary sequences.

Statistical Behavior	EPC Specification(standard)	Melia-Segui et. al.	Warbler	LAMED	J3Gen	Chen et. al.	AKARI-1 and AKARI-2
Probability of a single sequence (after analyzing 30 million 16-bit sequences)	$0.8 / 2^{16} < P(j) < 1.25 / 2^{16}$	$0.9 / 2^{16} < P(j) < 1.09 / 2^{16}$	$0.9409 / 2^{16} < P(j) < 1.0693 / 2^{16}$	$0.96 / 2^{16} < P(j) < 1.05 / 2^{16}$	$0.8 / 2^{16} < P(j) < 1.25 / 2^{16}$	$0.8 / 2^{16} < P(j) < 1.25 / 2^{16}$	$0.8 / 2^{16} < P(j) < 1.25 / 2^{16}$
Probability of simultaneously identical sequences (per ten thousand tags)	**shall be less than 0.1%**	**Almost 0 (zero)**	**approximately 2^{-45}**	**0.000157**	**0.0383 %**	**0.0026 %**	**less than 0.1%**
Probability of predicting a sequence (after 10 ms)	**shall not be predictable with a probability greater than 0.025%.**	**.000036 %**	2^{-16}	$2^{-11.77}$	**not be predictable with a probability greater than 0.025%.**	**not be predictable with a probability greater than 0.025%.**	**not be predictable with a probability greater than 0.025%.**

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

Table 5: Different lightweight pseudorandom number generator (PRNG) for EPC Class-1 Generation-2 (EPC C1 Gen2) RFID tags tested under NIST specification [23][24] [25][26].

According to NIST, the frequency test should be done first because it reveals the non-uniformity of a sequence. It gives the proportion of 0's and 1's for the entire sequence. This should be approximately equal for the truly random sequence. This is very useful in the case of the challenge-response mechanism between small battery operated devices. We can't use the complex data encryption methods or large hash functions for these devices.

Different lightweight pseudorandom number generator (PRNG) for EPC Class-1 Generation-2 (EPC C1 Gen2) RFID tags tested under NIST specification is given in TABLE 5.

Non-linearity may be achieved by using a non-linear Boolean function, filter generator, irregular clocking. Linear feedback shift registers can generate a pseudo-random sequence. Some attacks like fast algebraic attack and correlation attack can observe its sequence. Filter generator using a non-linear function or combination generator using a non-linear Boolean function can process the output sequence of an LFSR.

Several methods are in use to convert the output of an LFSR in a non-linear form, such as combination generators, clock-controlled generators, and Dynamic Linear Feedback Shift Registers. A control mechanism modifies the input to feedback function irregularly in order to protect it from several attacks. Some of the constructed DLFSRs are given in TABLE 6 along with the basic concept used in different DLFSR construction. Dynamic Linear Feedback Shift Register is another method in which the feedback polynomial changes dynamically at runtime.

Sl. No.	Author	Year	Basic concept used
1	Mita et. al.	2002	primitive polynomial
2	Mita et. al.	2006	decoder circuits and counter
3	Horant and Guinee	2006	clocking and polynomial switching time
4	Kiyomoto et. al.	2007	secondary LFSR as well
5	Molina – Rueda et. al.	2008	irreducible polynomial
6	Cid et. al.	2009	non-LFSR state
7	Snow 2.0	2010	dynamic number generator function
8	Colbert et. al.	2011	irreducible polynomial and hash function
9	Melui – Segui et. al.	2013	round robin scheme
10	Peninado et. al.	2014	two LFSR and a counter

Table 6: Basic concept used in different DLFSR construction

The general construction of a DLFSR is depicted in FIGURE 2. In a feedback shift register one bit is shifted to the right when needed and the new leftmost bit is calculated using the feedback function shift registers generates keys for ciphers.

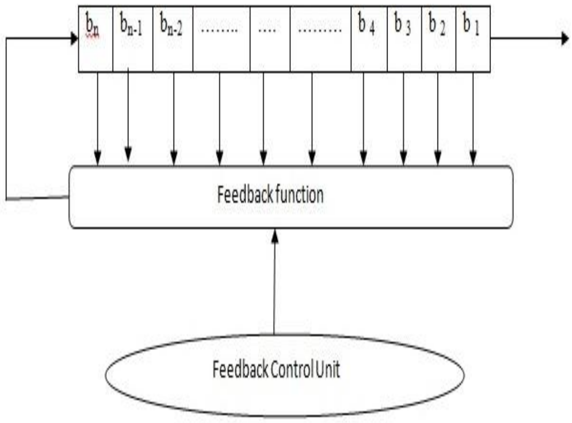

Fig 2: The general construction of a DLFSR.

Some of the criteria include period, linear complexity, and statistical measure. The statistical measure can be taken by using the Federal Information Processing Standard (FIPS) tests, Diehard suite, and the National Institute of Standard and Technology (NIST) statistical test suite [23].

IV. RESULT AND DISCUSSION

From the comparison table of different ciphers, it is clear that Pandaka top the list with the lowest gate equivalent of only 760 having 16-bit block size with 96-bit key size. Grain (80, 1) outperforms Trivium (80, 1) by 1130 gate equivalents. Trivium needs 2580 whereas Grain needs only 1450 gate equivalents. For different variants of different ciphers we can clearly see an increase in the requirement of gate equivalent with the increase in either key or block size.

Secure transmission needs strong security mechanism. While hashing, we need still lower gate equivalent algorithm for as large as 512-bit digest and that too should be capable of second preimage resistance. Linear feedback shift registers can generate a pseudo-random sequence. Some attacks like fast algebraic attack and correlation attack can observe its sequence. So we need some non-linear mechanism for the generation of yet another level of secure random numbers.

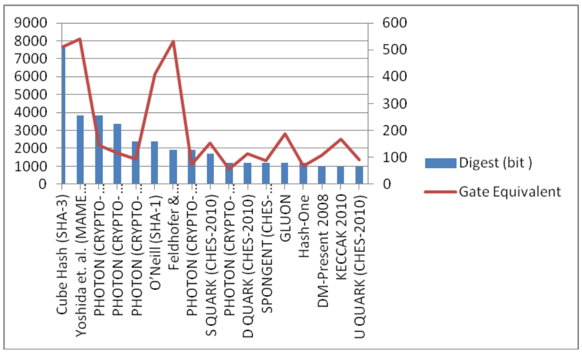

Fig 3: A comparison of required gate equivalent over the digest bit size.

We can see from FIGURE 3 that the size of the digest bit decreases in order to achieve the lower gate equivalent. Although many authors

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

favour the small bit digest it may largen the processing time. So, we need to work on large bit digests while lowering the gate equivalent in order to decrease the processing time as well. While from FIGURE 4 it is clear that the required ASIC for different hash functions is nearly the same.

From the comparison table of different random number generators, we can see a close fight among all of them because for every PRNGs probability of a single sequence (after analyzing 30 million 16-bit sequences) is fall in the range specified by NIST. Also, the probability of simultaneously identical sequences (per ten thousand tags) is as per NIST requirement.

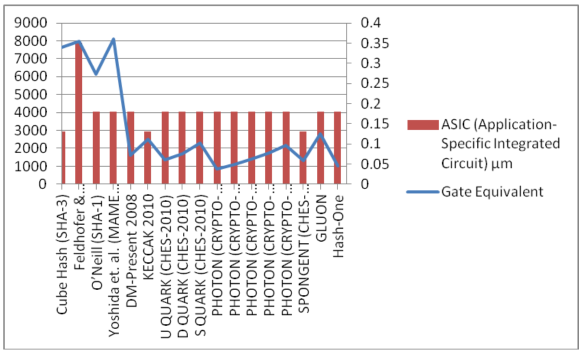

Fig 4: A comparison of required gate equivalent over the ASIC.

V.CONCLUSION AND FUTURE RESEARCH DIRECTION

We can see from the above facts that a lot of work had already been done in the field of security in transmission. As we move further towards an IoT arena, we are very much in need of lightweight security systems. These mechanisms should also be secure enough to tackle any type of security breach. For the same, researchers work in securing and light weighting HASH FUNCTIONS and RANDOM NUMBERS generation process. Many work and comparison of their results are given in this paper. From here we can progress towards the SPONGE based hash construction and DLFSR based random number generation. While SPONGE will give better preimage resistance on larger bit digest, DLFSR along with some set of LFSR will provide a high degree of randomness. Dynamic Linear Feedback Shift Register is the method in which the feedback polynomial changes dynamically at runtime.

REFERENCES

1. Katagi, M., & Moriai, S. (2008). Lightweight cryptography for the Internet of Things. *Sony Corporation*, 7–10. Retrieved from http://www.iab.org/wp-content/IAB-uploads/2011/03/Kaftan.pdf
2. Atzori, L., Iera, A., Morabito, G., & Diee, A. (2010). The Internet of Things : A survey. *Computer Networksxxx*. https://doi.org/10.1016/j.comnet.2010.05.010
3. Khan, R., Khan, S. U., Zaheer, R., & Khan, S. (2012). Future internet: The internet of things architecture, possible applications and key challenges. In *Proceedings - 10th International Conference on Frontiers of Information Technology, FIT 2012*. https://doi.org/10.1109/FIT.2012.53
4. Guest Editorial Special Issue on Internet of Things (IoT): Architecture, Protocols and Services. (2013). *IEEE SENSORS JOURNAL*, *13*(10). https://doi.org/10.1109/JSEN.2013.2274906
5. Singh, D., Tripathi, G., & Jara, A. J. (2014). A survey of Internet-of-Things: Future vision, architecture, challenges and services. In *2014 IEEE World Forum on Internet of Things, WF-IoT 2014*. https://doi.org/10.1109/WF-IoT.2014.6803174
6. Lucero, S. (2016). IoT platforms: enabling the Internet of Things, white paper. *IHS Technology*, *Whitepaper*(March), 1–19. Retrieved from https://cdn.ihs.com/www/pdf/enabling-IOT.pdf
7. Bröring, A., Schmid, S., Schindhelm, C. K., Khelil, A., Käbisch, S., Kramer, D., … Teniente, E. (2017). Enabling IoT Ecosystems through Platform Interoperability. *IEEE Software*, *34*(1), 54–61. https://doi.org/10.1109/MS.2017.2
8. Ngu, A. H., Gutierrez, M., Metsis, V., Nepal, S., & Sheng, Q. Z. (2017). IoT Middleware: A Survey on Issues and Enabling Technologies. *IEEE Internet of Things Journal*, *4*(1), 1–20. https://doi.org/10.1109/JIOT.2016.2615180
9. Weis, S., Sarma, S., Rivest, R., & Engels, D. (2004). Security and privacy aspects of low-cost radio frequency identification systems. *Security in Pervasive Computing*, *2802*, 201–212. https://doi.org/10.1007/978-3-540-39881-3_18
10. Molnar, D., & Wagner, D. (2004). Privacy and security in library RFID. *Proceedings of the 11th ACM Conference on Computer and Communications Security CCS 04*, (August), 210. https://doi.org/10.1145/1030083.1030112
11. Mohsen Nia, A., & Jha, N. K. (2016). A Comprehensive Study of Security of Internet-of-Things. *IEEE Transactions on Emerging Topics in Computing*. https://doi.org/10.1109/TETC.2016.2606384
12. EPC^{TM} Radio-Frequency Identity Protocols Generation-2 UHF RFID Specification for RFID Air Interface Protocol for Communications at 860 MHz – 960 MHz. (2013).
13. Singh, S., Pradip, ·, Sharma, K., Seo, ·, Moon, Y., Jong, ·, & Park, H. (n.d.). Advanced lightweight encryption algorithms for IoT devices: survey, challenges and solutions. *Journal of Ambient Intelligence and Humanized Computing*, *0*. https://doi.org/10.1007/s12652-017-0494-4
14. Taha, M., & Schaumont, P. (2014). Side-channel countermeasure for SHA-3 at almost-zero area overhead. In *Proceedings of the 2014 IEEE International Symposium on Hardware-Oriented Security and Trust, HOST 2014*. https://doi.org/10.1109/HST.2014.6855576
15. Mukundan, P. M., Manayankath, S., Srinivasan, C., & Sethumadhavan, M. (2016). Hash-One: a lightweight cryptographic hash function. https://doi.org/10.1049/iet-ifs.2015.0385
16. Aumasson, J. P., Henzen, L., Meier, W., & Naya-Plasencia, M. (2013). Quark: A lightweight hash. *Journal of Cryptology*. https://doi.org/10.1007/s00145-012-9125-6
17. Bogdanov, A., Kne??evi??, M., Leander, G., Toz, D., Varici, K., & Verbauwhede, I. (2011). Spongent: A lightweight hash function. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). https://doi.org/10.1007/978-3-642-23951-9_21
18. Guo, J., Peyrin, T., & Poschmann, A. (n.d.). The PHOTON Family of Lightweight Hash Functions.
19. 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 Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). https://doi.org/10.1007/978-3-642-31410-0_19
20. Kavun, E. B., & Yalcin, T. (2010). A lightweight implementation of Keccak hash function for radio-frequency identification applications. In Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). https://doi.org/10.1007/978-3-642-16822-2_20
21. Avoine, G., & Oechslin, P. (n.d.). A Scalable and Provably Secure Hash-Based RFID Protocol.
22. Rukhin, A., Soto, J., Nechvatal, J., Smid, M., Barker, E., Leigh, S., … Vo, S. (2010). A Statistical Test Suite for Random and Pseudorandom Number Generators for Cryptographic Applications.
23. Mandal, K., Fan, X., & Gong, G. (2012). Warbler: A lightweight pseudorandom number generator for EPC C1 Gen2 tags. In Cryptology and Information Security Series. https://doi.org/10.3233/978-1-61499-143-4-73
24. Peris-Lopez, P., Hernandez-Castro, J. C., Estevez-Tapiador, J. M., & Ribagorda, A. (2009). LAMED - A PRNG for EPC Class-1 Generation-2 RFID specification. Computer Standards and Interfaces.

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication

https://doi.org/10.1016/j.csi.2007.11.013
25. Melia-Segui, J., Garcia-Alfaro, J., & Herrera-Joancomarti, J. (2013). J3Gen: a PRNG for low-cost passive RFID. Sensors (Basel, Switzerland). https://doi.org/10.3390/s130303816
26. Melia-Segui, J., Garcia-Alfaro, J., & Herrera-Joancomarti, J. (n.d.). Analysis and Improvement of a Pseudorandom Number Generator for EPC Gen2 Tags.

AUTHORS PROFILE

Deena Nath Gupta is a Research Scholar in the department of Computer Science, Faculty of Natural Sciences, Jamia MIllia Islamia (Central University), New Delhi-110025, INDIA. He has done M.Tech in Computer Science and Engineering from Galgotia’s College of Engineering and Technology, Greater Noida, Gautam Buddha Nagar, Uttar Pradesh – 201306, INDIA and B.Tech in Computer Science and Engineering from ABES Engineering College, Ghaziabad, Uttar Pradesh – 201009, INDIA. He has an excellent academic background with academics and research experiences. He has published various research papers in the conferences of international/national repute. His research interest includes: Cryptography and cyber security, Internet of Things.

Rajendra Kumar is presently working as Assistant Professor in the Department of Computer Science, Faculty of Natural Sciences, Jamia Millia Islamia (Central University), New Delhi-110025, INDIA. He has an excellent academic background with a very sound academic and research experience. He has published various research papers in the conferences of international/national repute. His research interest includes: cyber security, Cloud Security and Privacy, Big Data Analytics, Data Mining, IoT, Software Security, Requirements Engineering, Security Policies and Standards, Software Engineering, Access control and Identity Management, Vulnerability Assessment etc.