IJCSE International Journal of Computer Sciences and Engineering Open Access
Research Paper Vol.-6, Issue-9, Sept. 2018 E-ISSN: 2347-2693

Dynamic S-box implementation in PRESENT Cipher

Kumar Anurupam

Computer Science and Engineering, RGUKT-IIIT, RGUKT, Nuzvid, India

**Corresponding Author: kumaranurupam@gmail.com, Tel.: +91-9885491191*

Available online at: www.ijcseonline.org

Accepted: 24/Sept/2018, Published: 30/Sept/2018

***Abstract*—** Lightweight cryptography is a very promising cryptographic technique which focuses on providing security to the smart devices in the IOT environment. The devices used in the IOT environment generate a large amount of data which can be very critical and sensitive to us. These IOT devices are generally very small and require low power to operate. So implementing strong cryptographic algorithms which need high computation is very difficult because of these limitations. Many lightweight ciphers are developed which focused on providing efficient encryption in these resource sensitive environment without taking much computational power. These ciphers are based on different approaches like AES, Feistel networks. In this paper it is proposed to make some change in the design of one such lightweight cipher i.e. PRESENT. The PRESENT cipher is based on the Substitution Permutation network and utilizes the S-box during the encryption. Here, the motivation is to improve the cipher technique and increase its efficiency by creating the dynamic S-boxes and comparing it with the static S-box on various factors.

***Keywords*—**IOT, AES, LightWeight Ciphers, PRESENT cipher, SP Network

I. INTRODUCTION

In modern times there is a lot of research and discussion going on in the field of IOT. Many new and efficient ways are developed to implement IOT architecture. Internet being the main backbone to implement this architecture is now available very easily. The broadband connectivity can be implemented wired or wireless with many smart devices utilizing it either of the way. Now the communication and interaction process is becoming very fast because of the ground breaking research done to reduce the component size and also to improve the efficiency of the chips and processors used in the appliances. Due to this growth, improvement is required in the area of adaptability and compatibility of the complex IP protocols and specialized sensor networks. The incompatibility issues occur because of the differing capability, communication, processing bandwidth and environment of IOT devices. Also focus is there to improve the confidentiality, integrity and availability of the data generated by these smart devices to maintain the privacy and secrecy [1].

The paper is organized in five sections. Section I focuses on the information about IOT, its usage and security concerns are discussed. The conventional encryption techniques are reviewed and their drawbacks are mentioned. Based on the drawbacks the need of lightweight ciphers are also explained. Section II covers the related work done by other people in the domain and the working of PRESENT cipher is briefly mentioned. Section III comprises the methodology used in the paper along with the proper explanation of the pseudo code used. Section IV shows the result of the approach used and the results are discussed with diagrams and graphs. Section V concludes the work and also in future how can we do more analysis on the approach used.

A. Usage of IOT

The first instance of smart device was observed at Carnegie Melon University where a coke machine was built which was used to keep the inventory of the drinks available and also whether the newly added drinks were cold or not. This proved to be a major invention in the area of smart devices. It motivated people to explore various areas and implement the concept in many new ways. Everyday many gadgets and appliances are getting connected to internet. Some of the areas in which the concept of IOT is being used nowadays in smart homes, wearable, smart city, transportation and many more [2] [17]. According to the Gartner report, the number of connected devices are expected to reach 11 billion units in the year 2018 and the forecast says that it will reach 20.4 billion in 2020.

According to the report published by IOT analytics which measured the percentage of IOT projects being carried out throughout the world, it was observed that the maximum number of projects were carried out in the area of smart city (367) followed by connected industry (265) and connected

building(193). The maximum percentage of smart city projects are being carried out at Europe. Americas (North and South) are doing very well in connected health, connected car and connected building. Smart agriculture is one field where Asia/Pacific region is doing better than European counterpart.

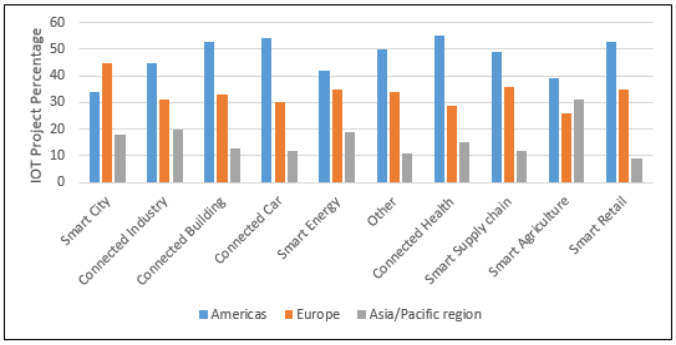

Figure 1. IOT Application Areas

B. *Security issues in IOT*

For any technology to be accepted by the people it is required to give a sense of confidence and security in it usage. To demonstrate this, the IOT architecture should give an assurance to the users that their data is secure and private and the chance of data compromise is minimal. But to attain this is a tough task.

There are many security challenges which is hampering the growth of IOT. These challenges occur due to various factors like capability of the device, capability of the device, insecure communication, authentication problem, vulnerability mismanagement and others [3][17]. All these factors plays a major role in hampering the security of IOT.

The main basis of the communication in smart devices is through wireless which poses many problems. The wireless network is not as secure as wired network which increases the chance of eavesdropping and other interception based attacks. With little effort an intruder can perform person in the middle attack and access all the communication going on between the devices. This compromises the working of IOT and raises a very important question on the security and privacy of data. So in order to make the communication secure we can make use of various encryption algorithm. But the IOT devices are having very less computing efficiency, therefore, it is very difficult to implement conventional encryption algorithms like AES which needs lots of computation for encrypting the data. The data stored in device and during the transit, both can be compromised.

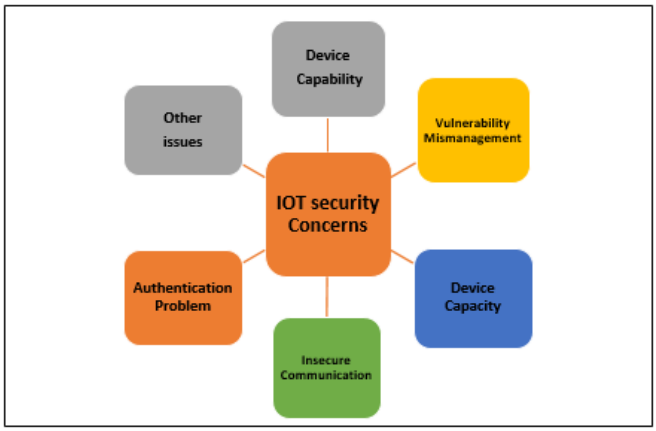

Figure 2. IOT Security Concerns

The other security concern is because of the memory limitation of the IOT devices. The smart devices being small is size are having very less memory. So the number of updates or patch to fix any vulnerability present in those devices is restricted to the memory capacity which ultimately exposes it to various kinds of attacks and risks.

The limitations of the smart devices makes it difficult to implement proper authentication of the users. The complex architecture of the IOT infrastructure creates a new kind of challenge to the developers to patch vulnerabilities in the network.

Due to these concerns related to security many companies are investing lots of money on improving the security of these smart devices. According to the Gartner report 2018 the money to be spent on IOT security will reach $1.5 billion in 2018 which is 28% increase from the year 2017 of $1.2 billion spending. It is also predicted that the investment will reach to $2.4 billion in year 2020. This shows how much people are thinking about IOT security by investing to improve the confidentiality, integrity and authentication related vulnerability.

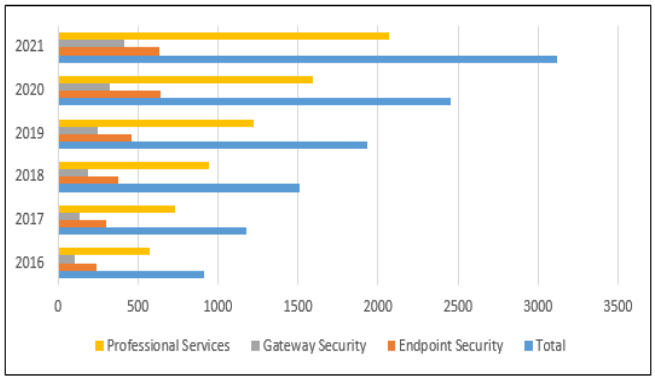

Figure 3. IOT Security Investments

C. Limitations of conventional encryption

There are many cryptographic encryption protocols which are there to facilitate an effective security encryption mechanism. Some of them takes a block of data and with the help of a key and series of rounds they convert the plain text to cipher text. Other variant is where the input is taken as a stream of bits and they are converted to cipher text with the help of an algorithm. Among these two, the former is more popular and many algorithms are using it. Some of the implementation of block cipher is DES, AES, etc. AES is the encryption algorithm which is commonly used in various application. It can be implemented in 3 variants on the basis of dividing the plaintext into blocks, the first where we take the block size as 128, the second with a block size of 192 and third where we can take block size as 256. Depending on the block size the number of rounds are decided. For block size of 128, 192 and 256 the number of rounds are 10, 12 and 14 respectively.

AES provides an efficient way to provide security. But security is not the only concern here. Being a very good encryption algorithm comes with a cost. In this case the cost is performance [4]. As the smart devices are very limited on resource therefore, it is very difficult to implement and maintain AES in these devices. The AES implementation requires lots of computation which are very complex and makes the device slow. Other important point here is that not all the devices requires same level of security. So, in that case those many number of rounds and calculation are not required.

D. Lightweight Cryptography

The limitations or drawbacks of the conventional encryption standard like AES in IOT environment gave rise to a new kind of encryption, known as Lightweight Cryptography. Many lightweight ciphers are now being used in smart devices using different approaches to optimize resources management. Some of them uses AES like architecture while others use Feistel network or some variant of Substitution Permutation Network (SPN). The ciphers which uses AES as a backbone are KLEIN [5], SKINNY [6] and LED [7]. Some of the lightweight ciphers based on Feistel networks are CLEFIA [8], LEA [9], SIMON and SPECK [10]. Other lightweight cipher based on SPN other than AES are PRESENT [11], PRIDE [12] and MANTIS [13]. The ciphers are created by keeping in mind the requirements of the IOT devices and network. They uses different algorithms to optimize the memory and space dependency.

II. RELATED WORK

The first instance when the lightweight cipher was thought to be developed initiated with eSTREAM [14]: the ECRYPT Stream Cipher Project. This project was focused on providing an efficient algorithm which took five long years to implement that is, it started in 2004 and completed in 2008. The eSTREAM was revised in September 2008, and currently contains seven stream ciphers named as HC-128, Rabbit, Salsa20/12, SOSEMANUK(128 bit ciphers), Grain v1, MICKEY 2.0, Trivium(80 bit ciphers) [15].

In 2007, Andrey Bogdanov, Lars R. Knudsen, Gregor Leander, ChristofPaar, Axel Poschmann, Matthew J. B. Robshaw, Yannick Seurin, and C. Vikkelsoe created a cipher which was very small in size and was named as PRESENT. The motivation to develop this cipher was to use it in devices which require high efficiency and low power consumption is desired. Next Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw proposed a new lightweight 64 bit block cipher LED [7] which uses PRESENT [11] cipher's S-boxes , Here they proposed new technique using four different key sizes, reasonable performance profile for software implementation. Wenling Wu and Lei Zhang proposed a new lightweight block cipher called LBlock [16]. It takes the data in a block size of 64-bit and the key as 80-bit. The motivation of this cipher was to focus on the defense against various attacks in the IOT environment ranging from cryptanalysis attacks to key dependent attacks. It was also developed by keeping in mind to implement it in both the hardware and software platforms. CLEFIA [8] is another kind of lightweight cipher developed by Taizo Shirai, Kyoji Shibutani, Toru Akishita, Shiho Moriai, and Tetsu Iwata by making use of feistel structure and worked on a block size of 128 bit.

PRESENT cipher is one of the most popular encryption algorithm which is used in RFID tags or the applications using the ubiquitous computing. Power and cost constraints were given high priority.

PRESENT is based on block encryption with size taken as 64 bits. It supports two key lengths of 80 and 128 bits. The working is based on the steps involving substitution and permutation which is also call as SP-network or SPN. It uses 31 rounds to complete the encryption.

The algorithm of PRESENT is shown in Figure 4 in which there are two main parts, iteration and expanding the key. The structure used is SPN. The iteration function is represented by F having three transformations: addRoundKey, S-box and Player. In 80 bit key version, the key is stored in a register K and the leftmost 64 bits are used in the iteration process. For other round the key register is rotated 61 positions to the left and the leftmost four bits are input in to the S-box of PRESENT and the output is XOR with the round counter to get the new key. After the iteration process is completed for the 64-bit plaintext P, the last round is XOR with the round key and the result is the 64-bit cipher C. The intermediate results after the transformation are

stored in the state. The complete process is represented in the following diagram.

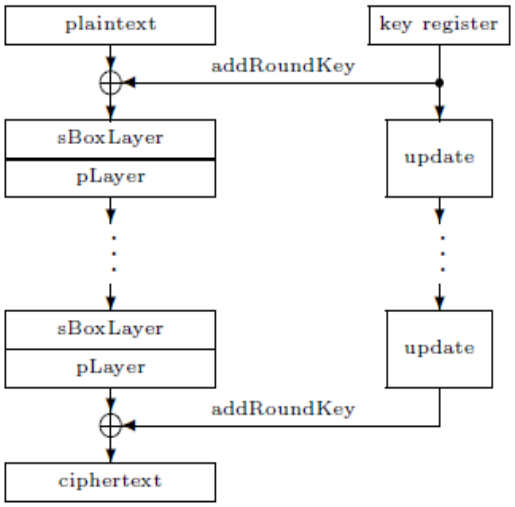

Figure 4. PRESENT Algorithm

Pseudo code to describe the PRESENT [11] cipher algorithm is as follows:
Generateroundkeys ()
For i=1 to 31 do
addRoundKey (STATE,Ki)
S-BoxLayer(STATE)
pLayer(STATE)
end for
addRoundKey(STATE,K32)

The encryption process is detailed as follows:

- addRoundKey: The input is of 64-bit which is XOR with round key.
- S-box: This is a non-linear 4-bit word substitution transformation, which is operating independently on each of the State 4-bit words.
- PLayer: It is a permutation transformation and works on the 64-bit State.

From the algorithm it is clear that PRESENT cipher uses 31 rounds. Xor operation is used in each of the 31 rounds to create a round key K_i for $1 \leq i \leq 32$, where the last key, K_{32} is used for a linear bitwise permutation and non-linear substitution layer and post-whitening. The non-linear substitution layer uses a single 4-bit S-box S which is used 16 times in parallel in each round of iteration.
After the completion of the three steps of addRoundKey, S-box transformation and PLayer the S/P network of PRESENT is as following.

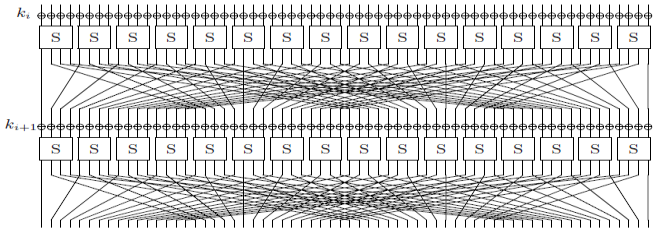

Figure 5. S/P network PRESENT

III. METHODOLOGY

In the paper a dynamic S-Box is created for each round. The dynamic nature of the S-Box makes the algorithm more versatile and reduce the chances of any kind of brute force attack which ultimately increases the level of security.

The S-Box which is used in the present cipher is static. The action of this box in hexadecimal format is mentioned below.

Table 1. S-Box PRESENT

x	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
$S[x]$	C	5	6	B	9	0	A	D	3	E	F	8	4	7	1	2

In the attempt to create Dynamic S-box a particular function is used to change the bit positions of S-box for each round. In each round a new S-box is created, so a total of 31 S-boxes are created. By this we can increase the security, unpredictability and efficiency of the cipher. The non-linear substitution layer uses a single 4-bit S-box in PRESENT. The implementation of such an S-box typically is much more compact than that of an 8-bit S-box.

A good performance of S-box should have good nonlinearity, differential uniformity, immune correlation and avalanche effect, and should avoid having fixed points or anti-fixed points. The genetic algorithm to design S-box gets the improved S-boxes (S-box S1 and S-box S2) with good diffusion rate, which solve the problem that the PRESENT S-box has anti-fixed point.

Table 2. Improved S-box S1

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(X)	C	B	6	5	9	0	A	D	3	E	F	8	4	7	1	2

Table 2. Improved S-box S1

X	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
S(X)	C	5	4	B	9	0	A	D	3	E	F	8	6	7	1	2

Pseudo code for dynamic Present Cipher is mentioned below

Generateroundkeys ()
For i=1 to 31 do
addRoundKey (STATE,Ki)
Dynamic_Sbox(Ki,Sbox)
S-BoxLayer(STATE)
pLayer(STATE)
end for
addRoundKey(STATE,K32)

Dynamic_Sbox method takes key for that particular round and Sbox. Dynamic S-box is generated by swapping the two positions i.e. (K_i mod 16) and (i mod 16) where i is round number and K_i is round key. Based on key value every time new S-box is created. Here we are having a total of 1632 number of combinations, so it is not that much easier to do cryptanalysis.

IV. RESULTS AND DISCUSSION

The performance of both the algorithm is compared by taking into consideration the varying input size and different processor speeds. So a graphical view of comparison is shown below to analyze both the comparisons.

The below figure compares the running time of the algorithms on the basis of different input size and block size in multiple of 8 bytes. Each BS (Block Size) mentioned in the figure represent a block of 8 starting from 8 and continuing till 80 bytes.

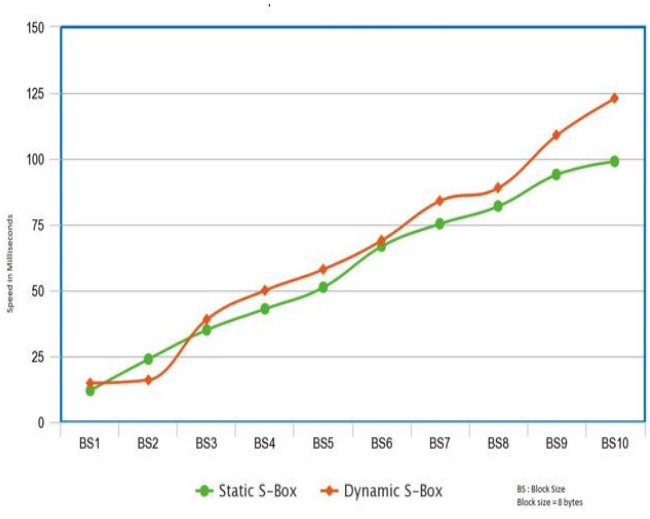

Figure 6. Running time comparison between two algorithms

The above graphs depicts that at initial block size of 8 bytes the running time is almost equal in both the implementation. But if we increase the block size as 80 bytes the dynamic implementation of S-box is working very efficiently than the static variant. Therefore, by analyzing the above comparison graph we can say that the speed of Dynamic S-box algorithm is high compared to the Static S-box algorithm.

The below graphs compare the running time of the static and dynamic S-box algorithm on the basis of different processor speed.

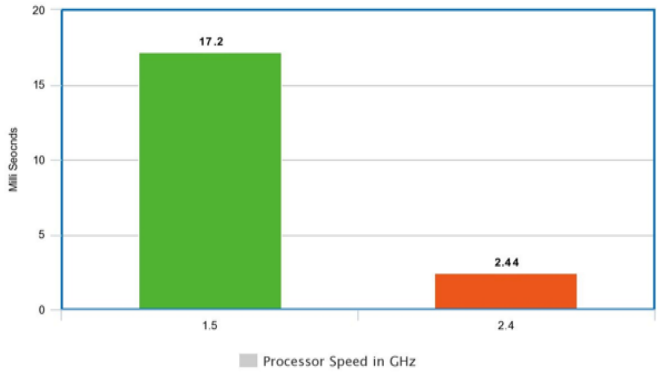

Figure 6. Running time comparison of static S-box PRESENT cipher on different processors

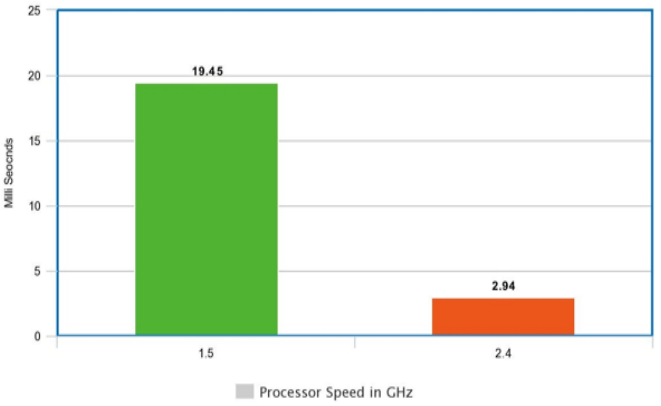

Figure 7. Running time comparison of dynamic S-Box PRESENT cipher on different processors

In the above graph two processors are taken, one of 1.5GHz and another of 2.4GHz and both the S-boxes are compared. By implementing the algorithms on these processors we can say that static S-box takes 17.2 milliseconds for encryption and while dynamic S-box is taking 19.45 milliseconds in 1.5GHz processor which is not very high comparing its counterpart. Similarly, in case of 2.4GHz static S-box takes 2.44 milliseconds for encryption and while dynamic S-box is taking 2.94 milliseconds in 1.5GHz processor which is again not very high.

On the basis of the previous diagram it is seen that the dynamic S-box is taking a little bit of extra time on the different processors but which is not very high and can be implemented without any trade-off.

V. CONCLUSION AND FUTURE SCOPE

In the coming times the usage of IOT is going to be very common in our daily life. So the work will continue to improve the efficiency of the ciphers used in the IOT environment. In the above study it is determined that by using the dynamic S-box in the PRESENT cipher we can increase the security level of PRESENT and also on various block size and input size it gives better performance than static S-box implementation. In terms of memory usage the dynamic S-box is slightly lagging behind the static S-box which shows that we can use it for IOT devices.

In the future work both the approaches can be compared on the basis of rate of diffusion and the standard deviation calculation. After the comparison more analysis can be done to get a more insight on the effectiveness of using dynamic S-box in the PRESENT cipher.

REFERENCES

[1] J. Lin, W. Yu, N. Zhang, X. Yang, H. Zhang and W. Zhao, "*A Survey on Internet of Things: Architecture, Enabling Technologies, Security and Privacy, and Applications*," in *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1125-1142, Oct. 2017.

[2] D. Makoshenko and I. Enkovich, "*IoT development: Discovering, enabling and validation of real life IoT scenarios*," *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, Valencia, pp. 159-164, 2017.

[3] M. Frustaci, P. Pace, G. Aloi and G. Fortino, "*Evaluating Critical Security Issues of the IoT World: Present and Future Challenges*," in *IEEE Internet of Things Journal*, vol. 5, no. 4, pp. 2483-2495, 2018.

[4] Sadiya Shakil and Vineet Singh, "*Security of Personal Data on Internet of Things Using AES*", International Journal of Computer Sciences and Engineering, Vol.4, Issue.6, pp.35-39, 2016.

[5] Zheng Gong, Svetla Nikova, and Yee Wei Law. 2011. "*KLEIN: a new family of lightweight block ciphers*". In Proceedings of the 7th international conference on RFID Security and Privacy (RFIDSec'11), Ari Juels and Christof Paar (Eds.). Springer-Verlag, Berlin, Heidelberg, pp. 1-18, 2011.

[6] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. 2016. "*The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS*". In Proceedings, Part II, of the 36th Annual International Cryptology Conference on Advances in Cryptology --- CRYPTO 2016 - Volume 9815, Matthew Robshaw and Jonathan Katz (Eds.), Vol. 9815. Springer-Verlag, Berlin, Heidelberg, pp. 123-153, 2016.

[7] Jian Guo, Thomas Peyrin, Axel Poschmann, and Matt Robshaw. 2011. "*The LED block cipher*". In Proceedings of the 13th international conference on Cryptographic hardware and embedded systems (CHES'11), Bart Preneel and Tsuyoshi Takagi (Eds.). Springer-Verlag, Berlin, Heidelberg, pp. 326-341, 2011.

[8] Shirai T., Shibutani K., Akishita 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, pp. 181-195, 2007.

[9] Deukjo Hong, Jung-Keun Lee, Dong-Chan Kim, Daesung Kwon, Kwon Ho Ryu, and Dong-Geon Lee. 2013. "*LEA: A 128-Bit Block Cipher for Fast Encryption on Common Processors"*. In Revised Selected Papers of the 14th International Workshop on Information Security Applications - Volume 8267 (WISA 2013), Yongdae Kim, Heejo Lee, and Adrian Perrig (Eds.), Vol. 8267. Springer-Verlag New York, Inc., New York, NY, USA, pp. 3-27, 2013.

[10] R. Beaulieu, S. Treatman-Clark, D. Shors, B. Weeks, J. Smith and L. Wingers, "*The SIMON and SPECK lightweight block ciphers*," 2015 52nd ACM/EDAC/IEEE Design Automation Conference (DAC), San Francisco, CA, pp. 1-6, 2015.

[11] Bogdanov A. et al. (2007) "*PRESENT: An Ultra-Lightweight Block Cipher*". In: Paillier P., Verbauwhede I. (eds) Cryptographic Hardware and Embedded Systems - CHES 2007. CHES 2007. Lecture Notes in Computer Science, vol 4727. Springer, Berlin, Heidelberg, pp. 450-466, 2007.

[12] Albrecht M.R., Driessen B., Kavun E.B., Leander G., Paar C., Yalçın T. (2014) "*Block Ciphers – Focus on the Linear Layer (feat. PRIDE)*". In: Garay J.A., Gennaro R. (eds) Advances in Cryptology – CRYPTO 2014. CRYPTO 2014. Lecture Notes in Computer Science, vol 8616. Springer, Berlin, Heidelberg, pp. 57-76, 2014.

[13] Christof Beierle, Jérémy Jean, Stefan Kölbl, Gregor Leander, Amir Moradi, Thomas Peyrin, Yu Sasaki, Pascal Sasdrich, and Siang Meng Sim. 2016. "*The SKINNY Family of Block Ciphers and Its Low-Latency Variant MANTIS"*. In Proceedings, Part II, of the 36th Annual International Cryptology Conference on Advances in Cryptology --- CRYPTO 2016 - Volume 9815, Matthew Robshaw and Jonathan Katz (Eds.), Vol. 9815. Springer-Verlag, Berlin, Heidelberg, pp. 123-153, 2016.

[14] Matthew Robshaw. 2008. "*The eSTREAM Project. In New Stream Cipher Design"s*, Matthew Robshaw and Olivier Billet (Eds.). Lecture Notes In Computer Science, Vol. 4986. Springer-Verlag, Berlin, Heidelberg, pp. 1-6, 2008.

[15] De Cannière C., Preneel B. (2008) "*Trivium*". In: Robshaw M., Billet O. (eds) New Stream Cipher Designs. Lecture Notes in Computer Science, vol 4986. Springer, Berlin, Heidelberg, pp. 244-266, 2008.

[16] Wu W., Zhang L. (2011) "*LBlock: A Lightweight Block Cipher*". In: Lopez J., Tsudik G. (eds) Applied Cryptography and Network Security. ACNS 2011. Lecture Notes in Computer Science, vol 6715. Springer, Berlin, Heidelberg, pp. 327-344, 2011.

[17] Deepanshu Mehta, "*Internet of Things: Applications and Challenges*", International Journal of Computer Sciences and Engineering, Vol.6, Issue.8, pp.289-293, 2018.

