

LEA-SIoT: Hardware Architecture of Lightweight Encryption Algorithm for Secure IoT on FPGA Platform

Bharathi R¹

Research Scholar
PRIST University, Thanjavur
Tamil Nadu, India

N. Parvatham²

Associate Professor
PRIST University, Thanjavur
Tamil Nadu, India

Abstract—The Internet of Things (IoT) is one of the emerging technology in today's world to connect billions of electronic devices and providing the data security to these electronic devices while transmission from the attacks is a big challenging task. These electronic devices are smaller and consume less power. The conventional security algorithms are complex with its computations and not suitable for IoT environments. In this article, the hardware architecture of the new Lightweight encryption algorithm (LEA) for the secured Internet of things (SIoT) is designed, which includes Encryption, decryption along with key generation process. The New LEA-SIoT is a hybrid combination of the Feistel networks and Substitution-permutation Network (SPN). The encryption/decryption architecture is the composition of Logical operations, substitution transformations, and swapping. The encryption/decryption process is designed for 64-bit data input and 64-bit key inputs. The key generation process is designed with the help of KHAZAD block cipher algorithm. The encryption and key generation process are executing in parallel with pipelined architecture with five rounds to improve the hardware and computational complexity in IoT systems. The LEA-SIoT is designed on the Xilinx platform and implemented on Artix-7 FPGA. The hardware constraints like area, power, and timing utilization are summarized. The Comparison of the LEA-SIoT with similar security algorithms is tabulated with improvements.

Keywords—IoT Devices; Security algorithm; Encryption; Decryption; Key generation; FPGA

I. INTRODUCTION

There are billions of devices connected on the internet, and a massive amount of data being generated from these devices and collect these data in IoT with authentication, service support, and privacy is a big challenging task. The future of the IoT is more in general to the public usage in-terms of smart homes, smart cities, virtual power plants, smart grids, intelligent transportation. To secure the data in the IoT environment, the lightweight security algorithms are suitable and essential because of less computational complexity. These lightweight algorithms are analyzed with a security level, hardware technology, throughput, latency, energy and power consumption, Memory utilization, and efficiency [1-3]. The IoT architecture is designed based on the layers used on the applications. The 3-layer architecture includes a physical, network, and application layer is used. The 4-layer

architecture includes data perception, heterogeneous network access, data management, and intelligent service layer. 5-layer IoT consists of Perception layer, network, processing layer, application, and business layer. The different attacks on IoT, which includes Denial of service (DoS), Man-in Middle, Wormhole, alteration, fabrication, and eavesdropping. The security challenges of IoT is security and data protection, authentication, privacy, access control, trust, and policy enforcement [4-6].

Section "A" discusses the background of the previous research works of Security algorithms for IoT applications followed by research gaps in section "B". Section II describes the Lightweight encryption algorithm (LEA) for the secured Internet of things (SIoT) with detailed hardware architecture. Section III explains the simulation results of the LEA-SIoT and analyze the hardware constraints of LEA-SIoT and comparison of the LEA-SIoT with other Security algorithms with improvements. Section IV concludes the overall work with improvements and future work.

A. The Background

This section discusses the existing work of different Security algorithms for IoT environments. The Goyal et al. [9] presents a Public key algorithm for low power IoT devices, which includes Elliptic Curve Diffie-Hellman (EC-DH) method. This EC-DH provides extended security with low power consumption for IoT gadgets with nominal processing speed and also compare with Diffie-Hellman (DH), and RSA algorithm with minor improvements. The Safi [10-11] describes the new hybrid encryption method to provide the security in IoT using public, private key along with digital signatures. The Advanced-encryption-standard (AES) is used for the public key, and NTRU is used for Private and digital signature. This hybrid method uses a software-based approach, which is not compactable with real-time hardware IoT environments. Khan et al. [12] present a performance analysis of different security algorithms like AES, SHA, and ECC using crypto++ library with C for small IoT applications and implemented on hardware Raspberry Pi-3. Landge et al. [13] describe the Message Digest (MD)-5 hashing algorithm for secure IoT using software platform. The sender and receiver hash information with execution time on server base is discussed. The Elliptic Curve scalar multiplication model is

presented by Venugopal et al. [14] for IoT security with smaller key sizes. The multiplier has two variants of karatsuba to improve chip utilization. Hajj et al. [15] present the analysis of the cryptographic algorithms on a hardware platform for IoT devices. The algorithms include symmetric, asymmetric, and, hashing, along with signature algorithms, are demonstrated on the Raspberry –Pi model. The hybrid encryption algorithm is described by Chandu et al. [16] which includes the AES algorithm for data transmission in the cloud and RSA algorithm encrypts the AES key which is used by the authorized user. Only AES part is performed by FPGA hardware, and the RSA part is done by Matlab environments. Liang et al. [17] present an authentication algorithm for identification of data under the IoT environment. The Hausdorff distance (HS) based algorithm is used for position selection, characteristic matching, and identification of data. Guruprasad et al. [18] present an analysis of security algorithms on FPGA platform which includes AES with different key sizes, Data encryption standard (DES), Triple DES (TDES), and Light encryption device (LED). Samir et al. [19] explain the lightweight hardware security algorithms for IoT on ASIC and FPGA platform. The hardware complexity in the IoT environment is resolved using authentication block ciphers. The dynamically reconfigurable security algorithm is presented by Yao et al. [20] for IoT, which includes AES and TDES algorithms are used to realize the dynamic switching in the reconfigurable partition, and core controller controls it. Tao et al. [21] explain about hardware-based block cipher with secured data collection for IoT based Healthcare, and KATAN security algorithm is used to provide crypto information from the patients and to doctor via a secured share cloud server. Ahmed et al. [22] presents the lightweight encryption algorithm for secured IoT on the Matlab platform and also implemented on 8-bit Microcontroller.

B. Research Gap

From the review of recent literature, it has been noticed that the amount of work carried on Security algorithms for IoT environment is based on software approaches and few on hardware-based approaches like simple microcontroller and raspberry-Pi. In the available existing hardware-based approaches carried most of them with conventional security algorithms. These existing algorithms are facing hardware complexity, performance degradation, and more chip area consumption with massive power consumption in the IoT environment. The existing lightweight encryption algorithms are uses more computational rounds for the confusion, which leads to more chip area and affect the system performance. These lightweight encryption algorithms face significant security challenges in the IoT environment. Thus there is a need for “*cost-effective new lightweight encryption algorithm for Secure IoT environment.*”.

II. PROPOSED WORK

In this section, the proposed Lightweight encryption algorithm is designed for secure IoT (LEA-SIoT) is explained with its hardware architectures.

The proposed algorithm provides a suitable simple and efficient Hardware architecture for real-time IoT environment

security. There are several block cipher algorithms with different network structures implemented with its own merits and demerits. In general, there are five different types of block ciphers available with different network structures namely Feistel network (FN), Substitution permutation networks (SPN), LFSR-based approach, Add-Rotate XOR (ARX) and hybrid combination approach.

In the proposed design, a hybrid approach is used with the combination of a Feistel network (FN) & SPN. The FN is having a major advantage of using almost similar encryption-decryption process. The SPN provides different rounds of substitution, which ensures the ciphertext is available in a pseudo-random manner. So the hybrid approach based architecture provides a new lightweight block cipher algorithm with efficient security and less computational complexity in IOT environment.

The lightweight encryption algorithm (LEA) for secured IoT (SIoT) is a symmetric key block-cipher (SKC) model which consist of 64-bit plain text and 64-bit key input. In general, most of the block cipher encryption process consists of many rounds of operations to keep the security process stronger. So each round of operation is designed based on some mathematics operations to create confusion. If the number of rounds increases, the system can provide better security with more resource utilization on a chip.

In the proposed design, only five rounds of operations are considered to improve the chip area and computational complexity in the IoT environment. The LEA-SIoT model mainly consists of encryption, decryption, and key generation process. Table I shows the notations with its functions used in the hardware architecture of the encryption, decryption, and key generation process. The detailed explanation of each process is discussed in the below section.

A. Key Generation Module

The complete data information, along with security is dependent on the key. If attackers know about the key, the data information will be lost. The encryption and decryption are processed with the same key input, which is having 64-bit. For each encryption-decryption round, the separate key generation input is provided to maintain the security. In this design, the Feistel-network based encryption algorithm is processed with five rounds using five different key inputs. The key generation provides the five unique keys to the encryption-decryption process.

TABLE. I. NOTATIONS

Symbol	Function
	XNOR
	XOR
	Register
	SBOX-Function
	Concatenation

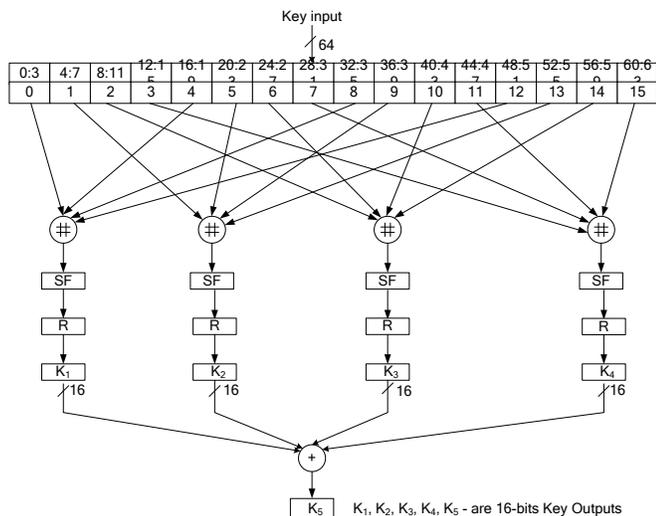


Fig. 1. Architecture of Key Generation Module.

The proposed algorithm provides a 64-bit key which is used to encrypt the 64-bit plain text data. The 64-bit key input (K_i) is considered from the user interest. The 64-bit key input is expanded in the key expansion block, and perform the substantial process to create the confusion for the given K_i to generate the five different keys. The key generation process mainly includes substantial process followed by concatenation and S-Box function (SF) to generate the five different keys like K_1 , K_2 , K_3 , K_4 , and K_5 are represented in Fig. 1. The key generation process is explained in the below steps as follows.

The 64-bit key input (K_i) is divided into 16 segments, each of 4-bits. Consider the 16-segments, perform initial substitution as per below equation (1).

$$K_{Si} = \sum_{j=1}^4 K_{I\ 4(j-1)+i} \quad (1)$$

Where $j = 1$ to 4 for each block creation with 4-bits and $i = 1$ to 4 for the first four rounds of key process. The single 16-bit block is generated with 4-segments. The first block K_{S1} contains $\{K_{I1}, K_{I7}, K_{I9}, K_{I13}\}$ as a first segment, $\{K_{I2}, K_{I6}, K_{I10}, K_{I14}\}$ as a second segment, $\{K_{I3}, K_{I7}, K_{I11}, K_{I15}\}$ as a third segment, and $\{K_{I4}, K_{I8}, K_{I12}, K_{I16}\}$ as the fourth segment for first round key generation process. Similarly, for K_{S2} , K_{S3} and K_{S4} will be calculated for second, third, and fourth blocks in key round process, respectively.

TABLE. II. P-TABLE FOR SBOX-FUNCTION

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
P(i)	3	F	E	0	5	4	B	C	D	A	9	6	7	8	2	1

TABLE. III. Q-TABLE FOR SBOX-FUNCTION

i	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Q(ii)	9	E	5	6	A	2	3	C	F	0	4	D	7	B	1	8

The 16-bit four blocks K_{S1} , K_{S2} , K_{S3} , and K_{S4} , are passed to S-Box function (SF) as represented in the equation (2).

$$K_i = SF(K_{Si}) \quad (2)$$

The S-Box function (SF) mainly contains P and Q tables; these tables have linear and non-linear transformations, which results in confusion and diffusion are represented in Fig. 2. The S-Box Function is taken from the Khazad block cipher [7][8]. The linear and non-linear transformations are represented by P and Q table is tabulated in Table II and Table III, respectively. The four S-Box function (SF) outputs are stored in four temporary registers (R) and the register outputs represented as first four key outputs, namely, K_1 , K_2 , K_3 , and K_4 .

To generate the 5th key K_5 , perform the XOR operation of the first four round keys (K_1 , K_2 , K_3 , and K_4), and it is represented in the below equation (3).

$$K_5 = \bigoplus_{i=1}^4 K_i \quad (3)$$

B. Encryption Module

The encryption process has 64-bit plain text input along with 5-round keys input, and the Hardware architecture of the encryption process is represented in Fig. 3. The encryption process mainly composed with logical operators (XOR, XNOR), substitution transformations and swapping along with pipelined registers in the architecture. The encryption process runs parallel with a key generation process with round keys to improve the latency of the system.

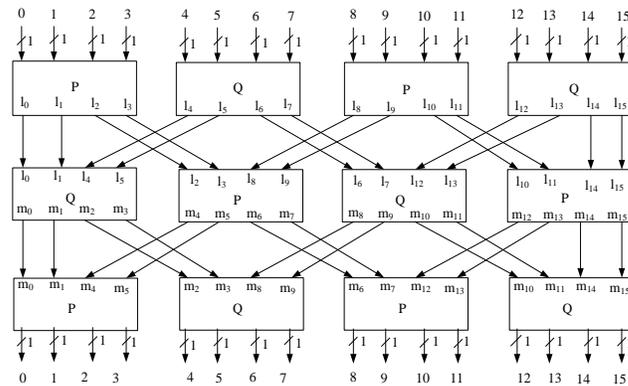


Fig. 2. Architecture of S-BOX Function (SF).

First, decompose the 64-bit plain text (P_T) into 4-blocks. Each block contains 16-bit input data. The plain text (P_T) has 4-blocks, $P_T [0:15]$, $P_T [16:31]$, $P_T [32:47]$ and $P_T [48:63]$. These 4-blocks are used to process the encryption operation. These 4-blocks input data are processed in each round with swapping to altering the originality of bits, and it is necessary to create the confusion and diffusion about ciphertext. The corresponding round key K_i from key generation is performed the bitwise- XNOR operation with $P_T [0:15]$ to generate the R_{11} and lk_1 in the left-hand side. Similarly with $P_T [48:63]$ to generate the R_{14} and rk_1 in the right-hand side, respectively. The XNOR outputs lk_1 and rk_1 are feed separately to S-Box function (SF) to generate the lf_1 and rf_1 as represented in Fig. 3.

The same key generation S-Box-function (SF) is used in the encryption process. To perform the XOR operation of lf_1 and rf_1 separately with swapped plain text $P_T [32:47]$ and $P_T [16:31]$ to generate the R_{12} and R_{13} outputs respectively. The R_{11} , R_{12} , R_{13} , and R_{14} are first round encryption outputs and stored in a pipelined register (R) to maintain the synchronization with key generation process. In general, the round operation outputs are expressed in the below equation (4).

$$P_{T i, j} \text{XNOR} K_i ; j = 1 \& 4$$

$$R_{i, j} = P_{T i, j+1} \oplus lf_i ; j = 2$$

$$P_{T i, j-1} \oplus rf_i ; j = 3$$
(4)

Where $i = 1$ to 4 to perform the second, third, fourth, and fifth round transformation outputs. The final ciphertext will be generated after the 5th round transformation, and are shown in equation (5).

$$C_T = R_{51} \# R_{52} \# R_{53} \# R_{54}$$
(5)

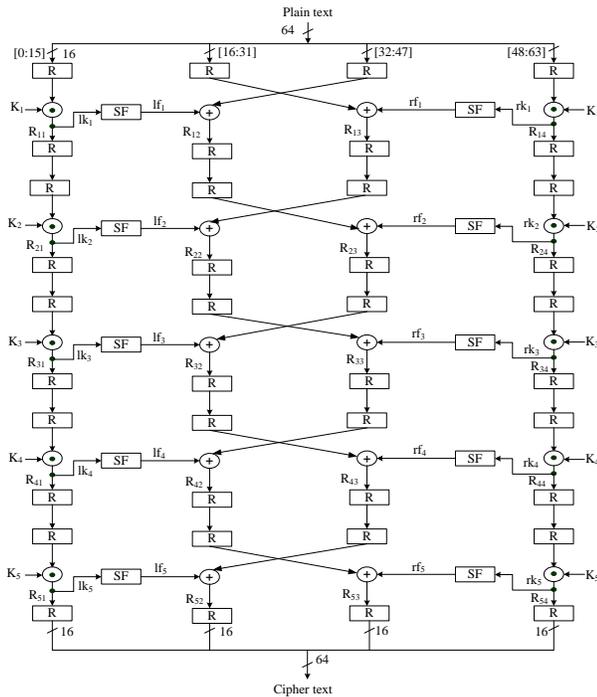


Fig. 3. Architecture of Encryption Module.

C. Decryption Module

The decryption process is almost similar to the encryption process with a few changes in the architecture, and it is represented in Fig. 4. The encryption process output 64-bit ciphertext as an input to the decryption process. First, decompose the 64-bit ciphertext (C_T) into 4-blocks. The 4-blocks are $C_T [0:15]$, $C_T [16:31]$, $C_T [32:47]$ and $C_T [48:63]$. These 4-blocks are used to process the decryption operation. The corresponding round key K_i from key generation is performed the bitwise- XNOR operation with $C_T [0:15]$ to generate the R_{11} and lk_1 in the left-hand side. Similarly with $C_T [48:63]$ to generate the R_{14} and rk_1 in the right-hand side, respectively. The $C_T [0:15]$ and $C_T [48:63]$ are feed separately to S-Box function (SF) to generate the lf_1 and rf_1 as represented in Fig. 4.

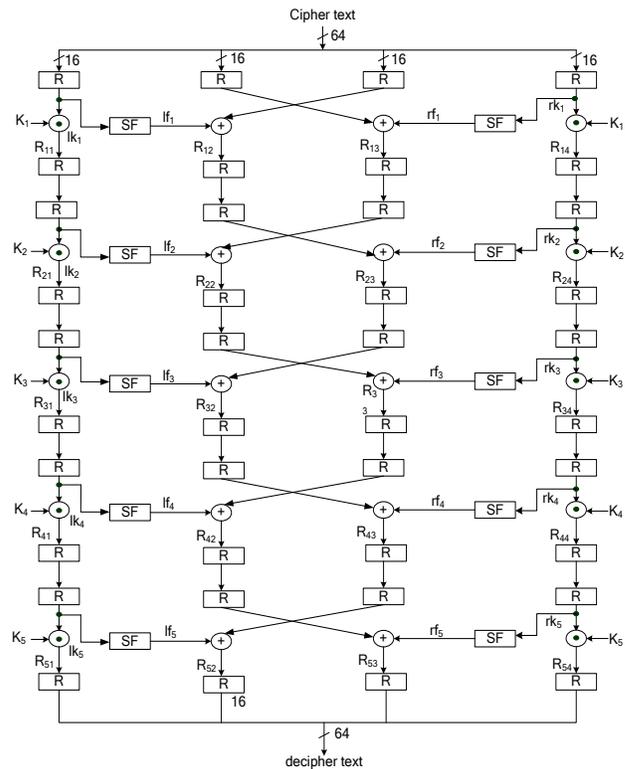


Fig. 4. Architecture of Decryption Module.

III. RESULTS AND ANALYSIS

The proposed Lightweight encryption algorithm for secure IoT (LEA-SIoT) results are analyzed in the below section. The Proposed work is designed using Xilinx ISE 14.7 tool using Verilog-HDL language, and modelsim 6.5f is used for simulation. The proposed LEA-SIoT is prototyped and implemented on the Artix-7 FPGA Platform by considering Device-XC7A100T-3CSG324.

The LEA-SIoT simulation results are represented in Fig. 5. The global clock (clk) signal is activated with the positive edge along with active low reset (rst). Set the 64-bit data input (Plain_text) to 64'haaaa_bbbb_cccc_dddd and 64-bit key input (key_in) to 64'haaaa_aaaa_aaaa_aaaa. As per Encryption and decryption process, 64-bit Encryption output

(Cipher_text) 64'h5555_a1dd_1c2f_2222 and decryption output (decipher_text) 64'haaaa_bbbb_cccc_ddd will be generated. The decipher output is the same as the plain text input with the delay of 2 clock cycle. The encryption process is performed in parallel with 5 rounds and similarly, for the decryption process. This simulation result indicates that the lightweight encryption algorithm works effectively with low latency for IOT environment.

The integration of both Encryption and decryption are instantiated as a single module named as LEA-SIoT is synthesized and after a place and route operation, the hardware resource utilization in terms of area, time, power and speed are summarized in Table IV. The LEA-SIoT utilizes less chip area in terms of 211 slice registers, 1578 Slice LUT's and 200 LUT-FF pairs on FPGA. The LEA-SIoT operated a maximum frequency of 249.457 MHz with a minimum period of 4.009ns on Artix-7 FPGA. The power consumption report is generated using the X-Power analyzer tool with FPGA system frequency of 100 MHz. The LEA-SIoT consumes the 0.249W total power with the inclusion of 0.167W dynamic power.

The proposed LEA-SIoT (only Encryption) is compared concerning previous security algorithms like AES, RSA and 3DES [19] in terms of performance parameters like Area (Slice LUTs) and Power (mWatts) at 10 MHz frequency are tabulated in Table V. The performance parameter results are analyzed on the Zynq 7000 series FPGA with device XC7Z020 for all the security algorithms.

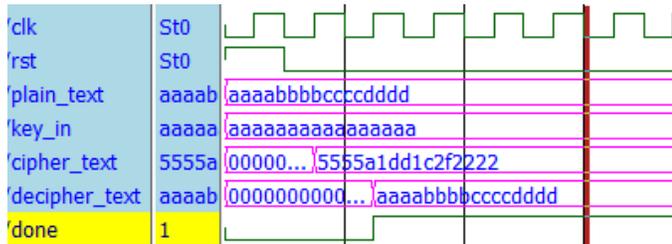


Fig. 5. Simulation Results of LEA-SIoT.

TABLE IV. HARDWARE RESOURCE SUMMARY OF PROPOSED LEA (ENCRYPTION AND DECRYPTION) FOR SECURE IOT

Resources	SIOT_ED
Area	
Number of Slice Registers	211
Number of Slice LUTs	1578
Number of fully used LUT-FF pairs	200
Time	
Minimum period (ns)	4.009
Maximum Frequency (MHz)	249.457
Power	
Dynamic Power (W)	0.167
Total Power(W)	0.249
Speed	
Throughput (Gbps)	7.982

TABLE V. SHOWS THE ENHANCEMENT OF AREA OVERHEAD (LUTs)

Design	Frequency (MHz)	Area (LUTs)	Power (mW)	FPGA Device
AES [19]	10	961	246	XC7Z020
RSA [19]	10	1178	255	XC7Z020
3DES [19]	10	1191	125	XC7Z020
Proposed Work	10	878	127	XC7Z020

The proposed design improves the area overhead (LUTs) around 10.49 % than AES, 25.56 % than RSA, and 26.28% than 3DES. The power consumption improved over around 48.37% than AES, and 50.19% than RSA at constant frequency 10 MHz (shown in Table V).

IV. CONCLUSION AND FUTURE WORK

The new Lightweight encryption algorithm (LEA) for the secured Internet of things (SIoT) is designed and implemented on Artix7 FPGA. The LEA-SIoT is a hybrid approach with Feistel networks and Substitution-permutation Network (SPN), for encryption/decryption. The LEA-SIoT reduces the computational and hardware complexity in IoT applications by executing the encryption/decryption and key generation process parallelly with pipeline architecture. The LEA-SIoT is simulated and synthesized on the Xilinx platform. The LEA-SIoT resource constraints like Area, time, Power, and speed utilizations are tabulated. The proposed algorithm for Secure IoT systems operate at 7.989 Gbps on Artix-7 FPGA and consume less power of 0.249W. The LEA-SIoT is compared with other security algorithms with improvements in the area (LUTs) and Power. The proposed LEA-SIoT (Encryption) model improves around 10.49% in Area and 48.37% in Power consumption than AES security algorithm. In the future, increase the number of encryption/decryption along with key generation rounds to strengthen the security with chip optimization, which suites in real time IoT applications.

REFERENCES

- [1] Hatzivasilis, George, Konstantinos Fysarakis, Ioannis Papaefstathiou, and Charalampos Manifavas. "A review of lightweight block ciphers." *Journal of Cryptographic Engineering* 8, no. 2 (2018): 141-184.
- [2] Surendran, Susha, Amira Nassef, and Babak D. Beheshti. "A survey of cryptographic algorithms for IoT devices." In 2018 IEEE Long Island Systems, Applications, and Technology Conference (LISAT), pp. 1-8. IEEE, 2018.
- [3] Sadeeq, Mohammed AM, Subhi RM Zeebaree, Riyadh Qashi, Sarkar Hasan Ahmed, and Karwan Jacksi. "Internet of Things Security: A Survey." In 2018 International Conference on Advanced Science and Engineering (ICOASE), pp. 162-166. IEEE, 2018.
- [4] Bhardwaj, Isha, Ajay Kumar, and Manu Bansal. "A review on lightweight cryptography algorithms for data security and authentication in IoTs." In 2017 4th International Conference on Signal Processing, Computing and Control (ISPCC), pp. 504-509. IEEE, 2017.
- [5] Pawar, Ankush B., and Shashikant Ghumbre. "A survey on IoT applications, security challenges, and countermeasures." In 2016 International Conference on Computing, Analytics and Security Trends (CAST), pp. 294-299. IEEE, 2016.
- [6] Naru, Effy Raja, Hemraj Saini, and Mukesh Sharma. "A recent review on lightweight cryptography in IoT." In 2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics, and Cloud)(I-SMAC), pp. 887-890. IEEE, 2017.
- [7] Barreto, P. S. L. M., and Vincent Rijmen. "The Khazad legacy-level block cipher." Primitive submitted to NESSIE 97 (2000).

- [8] Standaert, Francois-Xavier, Gael Rouvroy, Jean-Jacques Quisquater, and Jean-Didier Legat. "Efficient FPGA implementations of block ciphers KHAZAD and MISTY1." In Third NESSIE Workshop, pp. 6-7. 2002.
- [9] Goyal, Tarun Kumar, and Vineet Sahula. "Lightweight security algorithm for low power IoT devices." In 2016 International Conference on Advances in Computing, Communications, and Informatics (ICACCI), pp. 1725-1729. IEEE, 2016.
- [10] Safi, Amirhossein. "Improving the Security of the Internet of Things Using Encryption Algorithms." International Journal of Computer, Electrical, Automation, Control, and Information Engineering 11 (2017): 5.
- [11] Yousefi, Afsoon, and Seyed Mahdi Jameii. "Improving the security of internet of things using encryption algorithms." In 2017 International Conference on IoT and Application (ICIOT), pp. 1-5. IEEE, 2017.
- [12] Khan, Nuzhat, Nazmus Sakib, Ismot Jerin, Shaela Quader, and Amitabha Chakrabarty. "Performance analysis of security algorithms for IoT devices." In 2017 IEEE Region 10 Humanitarian Technology Conference (R10-HTC), pp. 130-133. IEEE, 2017.
- [13] Landge, Irfan A., and Hannan Satopay. "Secured IoT Through Hashing Using MD5." In 2018 Fourth International Conference on Advances in Electrical, Electronics, Information, Communication, and Bio-Informatics (AEEICB), pp. 1-5. IEEE, 2018.
- [14] Venugopal, Ellappan, and Tadesse Hailu. "FPGA Based Architecture of Elliptic Curve Scalar Multiplication for IOT." In 2018 Conference on Emerging Devices and Smart Systems (ICEDSS), pp. 178-182. IEEE, 2018.
- [15] El-Haii, Mohammed, Maroun Chamoun, Ahmad Fadlallah, and Ahmed Serhrouchni. "Analysis of Cryptographic Algorithms on IoT Hardware platforms." In 2018 2nd Cyber Security in Networking Conference (CSNet), pp. 1-5. IEEE, 2018.
- [16] Chandu, Y., KS Rakesh Kumar, Ninad Vivek Prabhukhanolkar, A. N. Anish and Sushma Rawal. "Design and implementation of hybrid encryption for the security of IOT data." In 2017 International Conference On Smart Technologies For Smart Nation (SmartTechCon), pp. 1228-1231. IEEE, 2017.
- [17] Huang, Yin, Wei Liang, Jing Long, Jianbo Xu, and Kuan-Ching Li. "A Novel Identity Authentication for FPGA Based IP Designs." In 2018 17th IEEE International Conference On Trust, Security And Privacy In Computing And Communications/12th IEEE International Conference On Big Data Science And Engineering (TrustCom/BigDataSE), pp. 1531-1536. IEEE, 2018.
- [18] Guruprasad, S. P., and B. S. Chandrasekar. "An evaluation framework for security algorithms performance realization on FPGA." In 2018 IEEE International Conference on Current Trends in Advanced Computing (ICCTAC), pp. 1-6. IEEE, 2018.
- [19] Samir, Nagham. "ASIC and FPGA Comparative Study for IoT Lightweight Hardware Security Algorithms." Ph.D. diss., Faculty of Engineering, Cairo University, Giza, 12613, Egypt, 2018.
- [20] Wang, Zhu, Yan Yao, Xiaojun Tong, Qinghua Luo, and Xiangyu Chen. "Dynamically Reconfigurable Encryption and Decryption System Design for the Internet of Things Information Security." Sensors 19, no. 1 (2019): 143.
- [21] Tao, Hai, Md Zakirul Alam Bhuiyan, Ahmed N. Abdalla, Mohammad Mehedi Hassan, Jasni Mohamad Zain, and Thayer Hayajneh. "Secured data collection with hardware-based ciphers for iot-based healthcare." IEEE Internet of Things Journal 6, no. 1 (2019): 410-420.
- [22] Usman, Muhammad, Irfan Ahmed, M. Imran Aslam, Shujaat Khan, and Usman Ali Shah. "SIT: a lightweight encryption algorithm for secure internet of things." arXiv preprint arXiv: 1704.08688 (2017).