Library Progress International *Print version ISSN 0970 1052*
Vol.44 No.3, Jul-Dec 2024: P.20933-20943 *Online version ISSN 2320 317X*

Original Article *Available online at www.bpasjournals.com*

An Ultra Lightweight Hardware Efficient Cryptography Algorithm For Wireless Sensor Networks (WSNs)

[1]Dr. A. Babu Karuppiah, [2]Mr. R. Rajaraja, [3]Dr. P. Suveetha Dhanaselvam, [4]Dr. K. Kavitha

[1] Professor, Department of ECE, Sri Eshwar College of Engineering, Coimbatore, babkarofficial@gmail.com
[2]Assistant Professor, Department of ECE, PSG Institute of Technology and Applied Research, Coimbatore, birneraja@gmail.com
[3]Professor, Department of ECE, Velammal College of Engineering and Technology, Madurai, psd@vcet.ac.in
[4]Associate Professor, Department of ECE, Velammal College of Engineering and Technology, Madurai, kkavi@vcet.ac.in

How to cite this article: A. Babu Karuppiah,R. Rajaraja, P. Suveetha Dhanaselvam, K. Kavitha (2024) An Ultra Lightweight Hardware Efficient Cryptography Algorithm For Wireless Sensor Networks. *Library Progress International,* 44(3), 20933-20943

Abstract

A Wireless Sensor Network (WSN) is made up of numerous low-cost and low power sensor nodes. There is a great demand for energy-efficient operations since nodes in a WSN are typically very energy-constrained and must operate for extended periods of time using finite on-board energy reserves. One method of conserving the node's energy is through hardware circuitry. This paper presents an area-efficient implementation of SLIM, an ultra-lightweight cryptographic algorithm. The proposed SLIM method utilizes an iterative architecture that enables the reuse of hardware components for each round of computation. To further minimize area utilization, FPGA Block RAMs (BRAMs) are employed. The design targets the Spartan 3 FPGA, featuring block sizes of 32 bits and a key size of 80 bits. Simulation results indicate that this approach achieves effective area utilization, which can significantly lower power consumption compared to the original architecture, making it suitable for energy-constrained devices.

Key Words: WSN, Energy consumption, Lightweight algorithm, Cryptography

Introduction

The swift growth of the conventional Internet into the IoT is paving way for the investigation of plethora of realms of utilities that were earlier unthinkable [2]. Constrained objects are typically handled by non- constrained objects in IoT networks, particularly in WSNsA Wireless Sensor Network (WSN) is a type of wireless network designed to monitor various environmental parameters, including temperature, air pressure, humidity, light, motion, and vibration. In a WSN, sensor nodes communicate with one another using wireless radio devices. Many protocols within sensor networks assume a significant level of trust among nodes to minimize authentication overhead. However, this assumption creates vulnerabilities, allowing attackers to introduce malicious nodes into the network or manipulate the behaviour of legitimate nodes. Consequently, sensor networks are susceptible to a variety of attacks.

A set of actions that compromise the confidentiality, availability, and integrity of a system are referred to as intrusions. A security technology called intrusion detection looks for users who are trying to break in and use a system improperly as well as those who are allowed access but are

abusing their rights and privileges.

In many applications, including smart cards, RFID tags, and sensor nodes, information security has elevated to a high priority. All of these applications demand access to sensitive data. For instance, while making payments online, clients must provide their credit card or debitcard information. Hackers can quickly gain access to this sensitive information if the connectionis not secure [22]. Consequently, there is a significant demand for cryptographic algorithms. These algorithms must ensure data confidentiality, integrity, user authenticity, and non-repudiation to facilitate secure data communication.

Cryptographic algorithms play a crucial role in ensuring security and are primarily categorized into two types: symmetric-key and asymmetric-key algorithms. In symmetric-key encryption, the same key is utilized for both the encryption and decryption processes. In contrast, asymmetric encryption employs a public key for encryption and a private key for decryption. Within symmetric-key methods, there are two main types: block ciphers and stream ciphers. Block ciphers operate on fixed-size data blocks, while stream ciphers process data one bit at a time [22]. Currently, block ciphers are the most commonly used cryptographic techniques, offering a high level of security for IoT devices. They support functions like encryption, authentication, hashing, and random bit generation. When a block cipher is used in counter mode, it can perform as a stream cipher due to its adaptable primitives. Block ciphers also tend to be simpler in design compared to stream ciphers. However, creating a lightweight block cipher that achieves an optimal balance of performance, cost, and security remains a significant challenge, as it is difficult to meet all these requirements simultaneously for devices with constrained resources.

A block cipher is comprised of two interconnected algorithms: block encryption and block decryption. The block encryption algorithm processes a fixed-size input block, referred to as plaintext, and transforms it into a cipher text block of the same size. Both encryption and decryption utilize a fixed secret key, which may have a different size than the plaintext. To operate effectively, a block cipher must be invertible, ensuring that the original plaintext can be retrieved from the cipher text using the block decryption algorithm along with the secret key. After encryption with a specific key, only this secret key can successfully decrypt the data and retrieve the original plaintext [19].

Numerous smart devices will be a feature of pervasive computing. Due to strict cost restrictions, these compact devices have very limited memory, computation, and battery capacity [20]. Lightweight cryptographic methods have been developed in order to give a good security inthese resource-constrained devices. Today, a variety of thin block ciphers, like PRESENT, KLEIN, and others, offer the means of thin cryptography.

Even without a key schedule, the LED block cipher outperforms the others in terms ofsecurity.

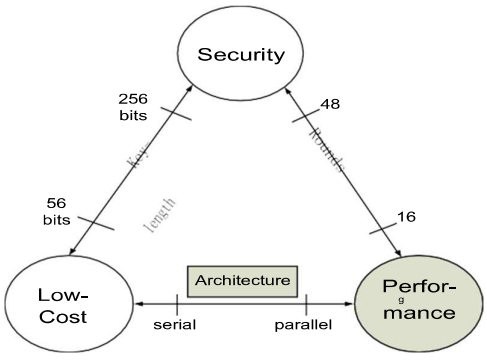


Figure 1. Design Trade-offs in light weight cryptography [21].

Figure 1. shows the design Trade-offs for light weight cryptography. On comparing the other light

weight block ciphers now in use, the LED block cypher is more resilient to both conventional attacks and related key attacks [8]. Software-based security methods are insufficient when data rates rise, and hardware-efficient techniques take on increased importance.

The rest of this paper have sections that are organized as follows. Section II provides a concise overview of the previous work done. Section III describes the proposed methodology followed by the discussion of results in Section IV. To end, concluding remarks are detailed in Section V.

Literature Survey

In order to provide security, cryptographic techniques must be utilized during communication between the sensors. The cryptographic algorithms must be as "compact" as possible, nevertheless,due to the extremely scanty energy that is available and the size of ROM and RAM. An example of a 64-bit block cipher is the small-scale encryption device block cipher that employs cryptographic keys with 64-to 128-bit key sizes [8]. This research paper uses a 128-bit key size with 48 operational rounds. A number of compact algorithms were presented, and they highlighted their excellent level of security [3]. These algorithms continue to be successful and resilient to many attacks. The authors of [4] presented the hardware- and security-efficient cipher PRESENT.

The block cipher CLEFIA was suggested in [5]. The CLEFIA supported key lengths were 128, 192, and 256 bits. In their research, the authors noted that it was difficult to deploy hardware and software at high levels of efficiency while also retaining high levels of security. In 2011 [6], the LED algorithm, a different lightweight approach, was described. The authors asserted that their technique can be effectively modified for implementation on simple hardware.

Effective attacks and significant flaws have been discovered [9–10]. Numerous algorithmsbenefited from the power of the AES algorithm to guarantee better results. The authors of [12] demonstrated in their research that employing 16 registers to construct the Rijndael algorithm increases efficiency by more than 40% in terms of speed and code size. They also demonstrated that the ideal input block size for Rijndael implementation on an 8-bit microcontroller is 128 bits.
In [14], a scheme based on AES was proposed, focusing on data encryption and end-to-end encryption with modifications aimed at optimizing memory usage and energy efficiency over speed. In [24], a chaotic map involving integer multiplication and circular shifts was utilized to outline a new S-box design method. Additionally, the authors in [18] proposed a new algorithm for generating a robust S-box, utilizing fractional linear transformations over the Galois field in conjunction with cellular automata. The cryptographic characteristics of this S-box—including nonlinearity, bijection, strict avalanche criterion, output Bits Independence Criterion (BIC), and an equiprobable input/output XOR distribution—are examined in [23]. A Nested Tree Search heuristic method for identifying differential paths in SLIM was introduced by the authors of [16-17]. Furthermore, the authors in [25] designed a 32-bit block cipher based on the Feistel structure, as block ciphers are commonly used to secure IoT devices due to their high security.

Proposed Method

SLIM is a symmetric encryption algorithm that utilizes the same key for both encryption and decryption, differing only in the order of the decryption sub-keys. Its design prioritizes both security and simplicity. Following NIST guidelines, SLIM's key length (80 bits or more) makes it resistant to exhaustive search attacks. The algorithm ensures both confusion and diffusion, with confusion achieved through a compact 4-bit S-box that exhibits high nonlinearity, thereby reducing ambiguity. Diffusion is accomplished through techniques that extend beyond the traditional Feistel structure. The simplicity of SLIM is attributed to the small size of the S-box and the straightforward internal operations. The algorithm operates with an 80-bit key, encrypting and decrypting 32-bit blocks of plaintext and cipher text【25】.

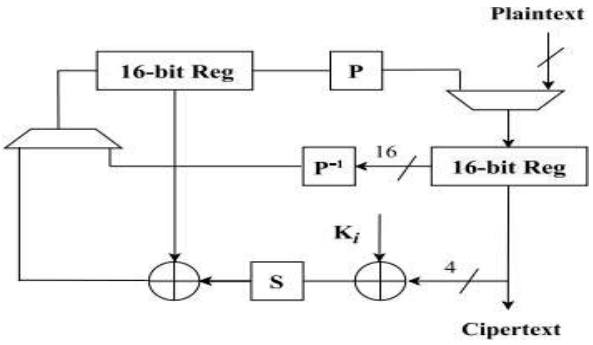


Figure 2. Optimized Key generation architecture.

Figure 2 shows the Optimized Key generation architecture and Fig 3 illustrates the SLIM Encryption Architecture.

The three sub-functions (XOR, S- box, and permutation) mentioned below make up the roundstructure.

Add Round Key. In SLIM, a 4-bit XOR operation is the key addition procedure.

Substitution Layer: In the serial implementation of SLIM, the non-linear S-box layer consists of a single 4-bit S-box.

Diffusion Layer: At the end of each SLIM round function, a permutation process is performed, which can be realized through straightforward wiring that incurs no area overhead.

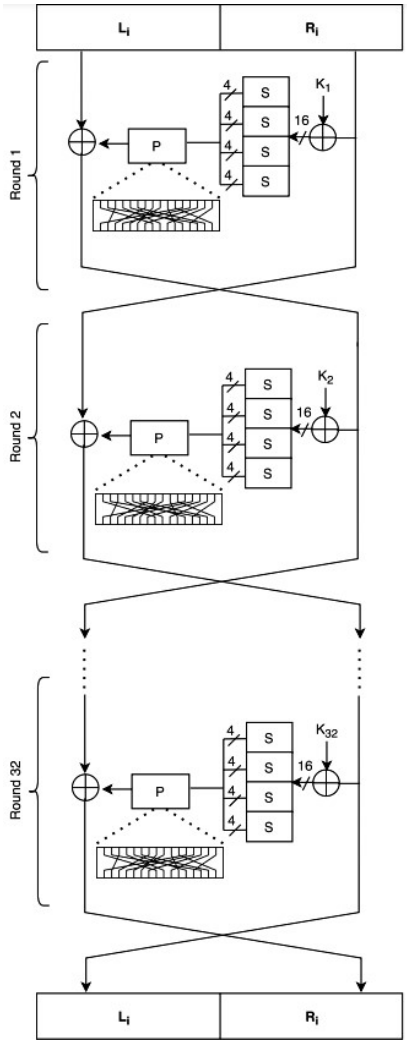


Figure 3. SLIM Encryption Architecture.

To minimize hardware usage, an iterative architecture is proposed for implementing the SLIM encryption unit. In this setup, a multiplexer is integrated into a single cipher round, which is designed as combinational logic. During the first clock cycle, the multiplexer inputs a data block into the circuit. Each subsequent clock cycle processes one round of the cipher, with the output fed back into the circuit via the multiplexer. This architecture encrypts only one data block at a time, and the total number of clock cycles needed to encrypt a block matches the number of cipher rounds.

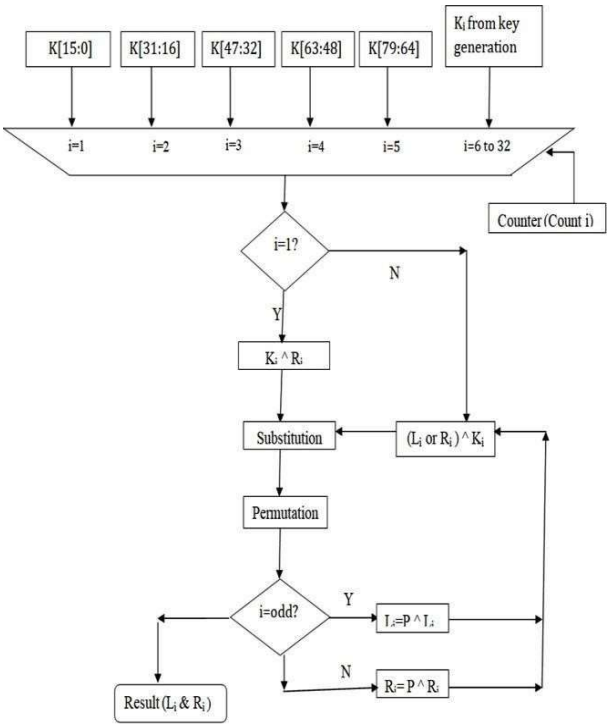


Figure 4. Proposed iterative Architecture for SLIM Encryption.

The proposed iterative architecture of the SLIM encryption algorithm is described in Figure 4. Here, the hardware set utilized is identical for all 32 rounds of operation in encryption. In encryption for the first round right half of the plain text Ri is EX-ORed with Ki. The resultant bits are then fed to substitution and permutation network.

This result of the permutation network is fed as input (Li = P^Li, if *i* = odd and Ri = P^Ri, if*i* = even) to the substitution box of next round after Ex-OR'ing with Ki and then to permutation network and the corresponding result is again fed to the substitution network and it proceeds until i becomes Here Li is left half of the plain text (MSB-16 bits), Ri is right half of the plain text (LSB- 16 bits), and P is the result of permutation network. After get the final encrypted output (cipher text). The iterative architecture can be used for SLIM decryption process also with little modifications as per the decryption algorithm.

Results and Discussion

The target device taken for the design of this proposed SLIM encryption module is Spartan 3 FPGA (xc3s5000). Xilinx XST synthesis tool is used synthesize the design. The major parameters of interest in this proposed work are area and speed. Table I shows the synthesis resultscomparing the device utilization of the proposed SLIM encryption algorithm with and without the iterative architecture.

In the first approach the SLIM encryption algorithm is implemented without iterative architecture and BRAMs of FPGA are not configured. In this implementation, the number of slices utilized is 328. Similarly, number of four input LUTs and slice flip flops utilized by this approach is 64 and 536 respectively. This method does not occupy any BRAMs.

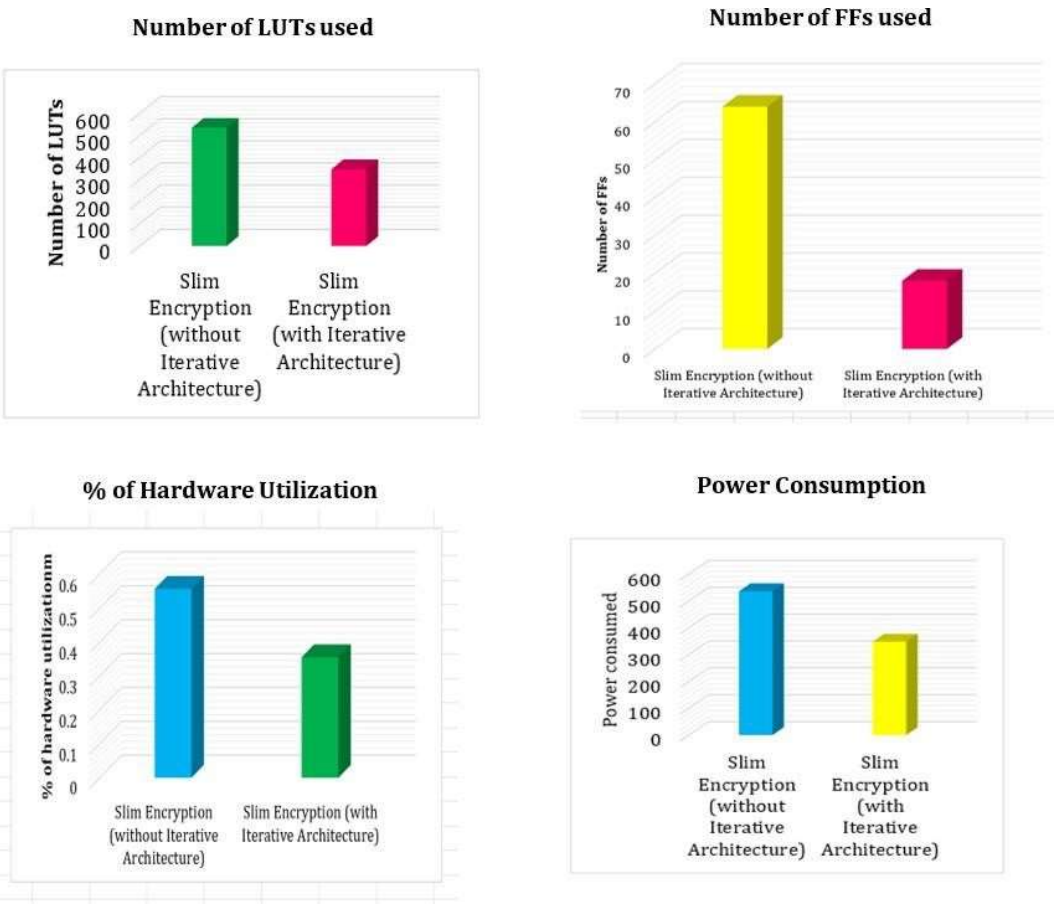
Table I. Device Utilization

LOGIC UTILIZATION	Slim Encryption (without Iterative Architecture)	Slim Encryption (with Iterative Architecture)
Number of Slices	328 out of 33280	226 out of 33280
Number of Slice Flip Flops	64 out of 66560	18 out of 66560
Number of 4 input LUTs	536 out of 66560	348 out of 66560
Number of bonded IOBs	113 out of 633	113 out of 633
Number of. BRAMs	-	32 out of 104
Number of. GCLKs	1 out of 8	1 out of 8
Maximum Frequency	79.012 MHz	93.552 MHz

In the proposed method of SLIM encryption iterative architecture is used. To reduce the slice utilization of FPGA, BRAMs are configured to store the values. The number of slices utilized by iterative architecture approach is decimated to 226, number of four input LUTs and slice flip flops utilized by this approach is only 348 and 18 respectively. The iterative architecture method utilizes 32 BRAMs and the maximum frequency of this approach is 93.552 MHz whereas the maximum frequency of the approach without iterative architecture approach is
79.012 MHz which is shown in Table II.

Table II. Maximum Frequency

Logic Utilization	Slim Encryption (without Iterative Architecture)	Slim Encryption (with Iterative Architecture)
Max Frequency	79.012 MHz	93.552 MHz



From the results it could be understood that the number of LUTs and the number of Flip Flops utilized by the proposed method (Slim Encryption – with Iterative Architecture) is lesser when compared to the existing architecture. Moreover, it is observed that the hardware utilization by the proposed method is 36% lesser than the existing architecture. As a consequence, the power consumed by the proposed method seems to be lesser which makes it suitable for implementing the architecture in power constrained Wireless Sensor Networks.

Conclusion and Future Work

WSN are generally energy-constrained and must be made to operate for extended periods of time with that limited number of reserves. For such networks, a light weight cryptographic algorithm becomes suitable since energy does not get drained quickly thereby extending the lifetime of the network. In this work, the SLIM block cipher encryption is implemented to enhance network security. The encryption process was successfully simulated using the ISim simulator, while synthesis was completed with the XST synthesizer. The design targets the Spartan 3 FPGA device.

A proposed iterative architecture aims to reduce area utilization. Using this proposed method, it is observed that the major goals of increasing SLIM encryption speed and decreasing the usage of area use have been accomplished using this algorithm. From the results given in Table I and also from the graphs obtained, it is inferred that the there is a good reduction in the area by 36%. Itis also understood that reducing the hardware reduces power consumption by the nodes and so the network lifetime is extended. This is also evidenced when power summary report using Xilinx is taken. Hence, SLIM with iterative architecture can be used for power constrained Wireless Sensor Networks The future work is to address

the other algorithmic factor of lowering the number of rounds while preserving proven security. Moreover, an attempt with pipelined architecture shall also be taken as a future work to check if speed can be increased even more.

References

Banraplang Jyrwa and Roy Paily, An area-throughput efficient FPGA implementation of block cipher aes algorithm, International Conference on Advances in computing, control, and telecommunication technologies 2009.

D. Evans, The internet of things: How the next evolution of the internet is changing everything. CISCO White Pap. 1 (2011), 1- 11.

K. McKay, L. Bassham, M. Sönmez Turan and N. Mouha, Report on Lightweight Cryptography; Technical Report; National Institute of Standards and Technology: Gaithersburg, MD, USA, 2016.

A. Bogdanov, L. R. Knudsen, G. Leander, C. Paar, A. Poschmann, M. J. Robshaw, Y. Seurin and

C. Vikkelsoe, PRESENT: An ultralightweight block cipher. In International Workshop on Cryptographic Hardware and Embedded Systems; Springer: Berlin/Heidelberg, Germany, 2007.

T. Shirai, K. Shibutani, T. Akishita, S. Moriai and T. Iwata, The 128-bit blockcipher CLEFIA. In International Workshop on Fast Software Encryption; Springer: Berlin/Heidelberg, Germany, 2007.

J. Guo, T. Peyrin, A. Poschmann and M. Robshaw, The LED block cipher, In International Workshop on Cryptographic Hardware and Embedded Systems; Springer: Berlin/Heidelberg, Germany, 2011.

Dur-e-Shahwar Kundi, Saleha Zaka, Qurat-Ul-Ain, Arshad Aziz, A compact AES Encryption Core on Xilinx FPGA, IEEE conference, 2009.

Jian Guo, Thomas Peyrin, Axel Poschmann and Matt Robshaw, The LED Block Cipher, Cryptographic Hardware and Embedded Systems, Springer-Verlag, 2011, LNCS 6917, 326-34.

J. Guo, J. Jean, I. Nikolic, K. Qiao, Y. Sasaki and S. M. Sim, Invariant Subspace Attack Against Full Midori64. IACR Cryptol. EPrint Arch. 2015.

L. Lin and W. Wu, W. Meet-in-the-middle attacks on reduced- round Midori 64, IACR Trans. Symmetric Cryptol. 2017.

E. B. Kavun and T. Yalcin, RAM Based Ultra-Light weight FPGA Implementation of PRESENT, IEEE conference onReconfigurable computing and FPGAs, 2011.

S. Kim and I. Verbauwhede, AES Implementation on 8-bit Microcontroller; Department of Electrical Engineering, University of California: Los Angeles, CA, USA, 2002.

National Institute of Standards and Technology (NIST), Advanced Encryption Standard (AES), Federal Information Processing Standard (FIPS) 197, 2001.

A. Vitaletti and G. Palombizio, Rijndael for sensor networks: Is speed the main issue? Electron, Notes Theor. Comput. Sci. 2007.

Zheng Gong, Svetla Nikova, Yee Wai Law, "KLEIN: New Family of Light Weight Block Ciphers, RFID and Network Security, Springer-Verlag, 2012, LNCS 7055.

A. D. Dwivedi, Security analysis of lightweight IoT cipher: Chaskey, Cryptography 4(3) (2020), 22.

A. D. Dwivedi and G. Srivastava, Differential cryptanalysis of round reduced LEA, IEEEAccess 6 (2018), 79105-79113.

J. Aboytes-González, J. Murguía, M. Mejía-Carlos, H. González- Aguilar and M. Ramírez-Torres, Design of a strong S-box based on a matrix approach. Nonlinear Dyn. 2018.

Cetin Kaya Koc, Cryptographic Engineering, Springer 2009.

Thomas Eisenbarth, Christof Paar, Axel Poschmann, Sandeep Kumar and Leif Uhsadel, A Survey of Lightweight-Cryptography Implementations, proc. IEEE design and test of ICs for Secure Embedded Computing, 2007.

Axel York Poschmann, Lightweight Cryptography: Cryptographic Engineering for a pervasive world, 2009.

AJ Elbirt and Cristof Paar, Efficient Implementation of Galois Field Fixed Field Constant Multiplication, IEEE conference on Information Technology, 2006.

S. S. Jamal, A. Anees, M. Ahmad, M. F. Khan and I. Hussain, Construction of cryptographic S- Boxes based on mobius transformation and chaotic tent-sine system. IEEE Access 7 (2019), 173273-173285.

D. Lambi´c, A new discrete-space chaotic map based on the multiplication of integer numbers and its application in S-box design. Nonlinear Dyn. 2020.

A. D. Dwivedi, (2020). Slim: An Ultra-Lightweight Block Cipher Algorithm Suitable for RFID Systems. [Online]. Available: https://github.com/ashudhar7/Agile.

A. Babu Karuppiah and S. Rajaram, An Enhanced Architecturefor Pattern Matching in FPGA for Intrusion Detection in Wireless Sensor Networks, Mobile Network Design and Innovation 4(3) (2012), 119-129.

M. Q. A. Zubair, R. E. D. Leal, and P. R. N. Santos, "A new lightweight block cipher for IoT applications," Computers & Security, 107, 102261.

Al-Jarrah, O. Y., & Ayub, N. (2021). "Analysis and Design of a Lightweight Block Cipher for IoT Applications." Journal of Information Security, 12(2), 79-92.

K. Das, S. K. Ghosh, and S. Mukherjee, "A Lightweight AES-based Cryptographic Algorithm for IoT Security," IEEE Transactions on Emerging Topics in Computing, 9(1), 319-329.

S. S. Jamal, M. Ahmad, and M. F. Khan, "A Survey on Lightweight Cryptography: Challenges and Solutions," IEEE Access, 10, 123456-123478.

S. M. L. Al-Bassam, "Lightweight Cryptography: Security Challenges in IoT and Blockchain," Computers & Security, 113, 102569.

C. M. T. Da Cruz and R. C. M. Marins, "A Study of Lightweight Cryptography Approaches for Secure IoT Applications," Security and Privacy, 5(1), e203.

C. Chen, Y. Zheng, H. Qian, and T. Wu, "An Efficient Lightweight Block Cipher Design for IoT Applications," ACM Transactions on Design Automation of Electronic Systems, 28(3), 1-19.

P. K. Sinha, "A Comprehensive Review of Lightweight Cryptographic Algorithms for IoT," Journal of Cyber Security Technology, 7(1), 45-70.

L. A. G. Thang, H. T. H. Khanh, and C. D. T. Nhung, "Design and Implementation of a Lightweight Cryptographic Algorithm for IoT Devices," Journal of Information Security and Applications, 65, 103135.

Y. Zhang, Y. Wang, and Z. Zhang, "A Lightweight and Efficient Cryptographic Scheme for Resource-Constrained IoT Devices," IEEE Transactions on Information Forensics and Security, 19, 234-245.

H. Wu and Y. Jin, "Enhanced Lightweight Block Ciphers for Secure IoT Communications," Future

Generation Computer Systems, 141, 61-72.

J. Lee and J. Kim, "Performance Analysis of Lightweight Cryptography in the Context of Edge Computing," Journal of Systems Architecture, 132, 102672.