**Journal of Information Systems Engineering and Management**
2025, 10(13s)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

**Research Article**

# SafeLink: End to End Privacy & Integrity for Data Transfer in IoT Applications

Neve Rahul P.[1], Vijaykumar Yele[2], Namdeo Badhe[3], Komal Madhukar Dhule[4]
[1*]*Associate Professor, IT department, Thakur College of Engineering and Technology. University of Mumbai, Maharashtra, India. rahulneve@gmail.com , orcid id : https://orcid.org/0009-0008-2800-929X*
[2*]*Assistant Professor, IT department, Thakur College of Engineering and Technology. University of Mumbai, Maharashtra, India. vijaypyele@gmail.com*
[3*]*Associate Professor, IT department, Thakur College of Engineering and Technology. University of Mumbai, Maharashtra, India. namdeob.badhe@thakureducation.org*
[1*]*Assistant Professor, IT department, Thakur College of Engineering and Technology. University of Mumbai, Maharashtra, India. komal.dhule@tcetmumbai.in*

**ARTICLE INFO**

Received: 01 Dec 2024

Revised: 15 Jan 2025

Accepted: 30 Jan 2025

**ABSTRACT**

In an era dominated by the Internet of Things (IoT) and ubiquitous connectivity, the demand for securing sensitive data, especially in surveillance systems, is paramount. This paper introduces a novel approach to enhancing security through the deployment of a lightweight cryptographic algorithm. The algorithm is specifically designed for resource-constrained environments, exemplified by its successful implementation on the Raspberry Pi platform.

The proposed cryptographic solution not only ensures the confidentiality of transmitted data but also addresses the resource limitations inherent in edge devices. Focusing on image encryption in surveillance systems, our hybrid lightweight cryptographic algorithm operates seamlessly on the Raspberry Pi, demonstrating its adaptability to low-resource environments.

The process begins with the capture of images, followed by their secure storage and processing on the Raspberry Pi. Subsequently, the images are encrypted, and the encrypted files are transmitted over the network. Notably, the security is fortified by incorporating a hash value that accompanies the encrypted file during transmission. At the receiving end, the integrity of the received file is verified by comparing the transmitted hash value with the computed hash value, allowing for the rejection of any tampered or compromised data.

This research contributes a holistic solution to the challenges posed by resource constraints in edge computing devices, offering a secure and efficient method for image transmission in surveillance systems. The presented hybrid lightweight cryptographic algorithm not only fortifies the confidentiality of data but also ensures the integrity of transmitted information, making it a valuable addition to the evolving landscape of secure IoT applications.

Keywords: End to End privacy, Cipher, decryption, IoT devices, security, resource constraint, attack analysis, lightweight cryptography.

## INTRODUCTION

IoT-kind devices with limited resources are unable to deal with traditional cryptographic algorithms due to the well-established constraints like low power, less memory and limited capacity as compared to the high - end processor of full-fledge devices, necessitating investigation to adopt lightweight cryptographic algorithm to achieve the requirements of resource constrained devices.Now a days there are many smart applications and system which is based on IoT resource constrained devices. In such system data is begin process on small controller like raspberry pi and further send to another device through communication network. To provide security for such devices in terms of confidentiality, traditional cryptographic algorithms doesn't work The increasing prevalence of surveillance systems in our interconnected world necessitates robust security measures to safeguard the confidentiality and integrity of the captured data. In response to the challenges posed by resource-constrained environments, recent advancements

in cryptographic research have witnessed a surge in the development and deployment of lightweight cryptographic algorithms tailored for specific applications. This technical introduction delves into the utilization of lightweight cryptography as a pivotal component in fortifying the security of surveillance systems, drawing insights from recent standard research papers in the field. Traditional cryptographic methods, while proven effective, often face hurdles when implemented in scenarios characterized by limited computational resources, such as edge devices in surveillance systems. Recognizing this, recent research endeavors have focused on the design and optimization of cryptographic algorithms that strike a delicate balance between robust security and efficiency.

In the context of surveillance, the demand for real-time image processing and secure transmission presents unique challenges. Standard research papers in lightweight cryptography have articulated innovative approaches to address these challenges, often leveraging specialized algorithms fine-tuned for low-resource environments. These algorithms are designed to operate seamlessly on edge devices like the Raspberry Pi, ensuring that the security measures do not compromise the efficiency and functionality of the surveillance system.[1][2][3]

One of the key breakthroughs in recent research papers involves the development of hybrid lightweight cryptographic algorithms. These algorithms not only excel in securing data but also take into account the limitations imposed by the processing capabilities of edge devices. By tailoring cryptographic solutions specifically for resource-constrained environments, researchers aim to provide a robust and scalable security framework for surveillance systems.This technical introduction sets the stage for a comprehensive exploration of the recent strides in lightweight cryptography and its application in securing surveillance systems. By amalgamating insights from standard research papers, we embark on a journey to understand the intricacies of these innovative cryptographic solutions, their impact on the surveillance landscape, and the potential they hold for shaping the future of secure data transmission in real-time surveillance scenarios. [4].This paper introduces a use of hybrid lightweight cryptographic algorithm, named Salted-HybridSIMONSPECKey, specifically designed for securing surveillance systems with resource-constrained devices.

The proposed algorithm combines the round function of the SIMON block cipher with the key scheduling technique of the SPECK algorithm, introducing a cryptographic random salt for added security. The design aims to optimize execution time, memory consumption, and energy efficiency, crucial aspects for devices like IoT sensors and wearables. The research includes a thorough attack resistance analysis, focusing on the algorithm's vulnerability to various cryptographic threats.

## EARLIER WORK

### LIGHTWEIGHT SIMON

*A. Encryption Process in SIMON*

Simon Encryption Steps (for a 64-bit block with a 128-bit key):[5][6]

1.Initialization:

i) Input: Plaintext (P), Key (K)

ii) Divide the 128-bit key into two halves: K1 and K2.

iii) Initialize the state: X = P.

2.Rounds (Repeat for a fixed number of rounds):

For each round (from 1 to the desired number of rounds):

i)Compute the round key:RK = K1 ⊕ (K2 <<< 3) ⊕ (K2 >>> 4)

(where <<< denotes left rotation and >>> denotes right rotation).

ii) Apply the round function: X = F(X) ⊕ RK.

iii) Update the key halves: (K1, K2) = (K2, K1 ⊕ (K2 <<< 1) ⊕ (K2 >>> 8)).

3.Final Round:

i.Compute the final round key: RK_final = K1 ⊕ (K2 <<< 3) ⊕ (K2 >>> 4).

ii.Encrypt the plaintext: Ciphertext © = F(X) ⊕ RK_final.

B. Decryption Process in SIMON

(for a 64-bit block with a 128-bit key):

1. Initialization:

Input: Ciphertext ©, Key (K)

Divide the 128-bit key into two halves: K1 and K2.

Initialize the state: X = C.

2. Rounds (Repeat for the same number of rounds as encryption):

For each round (from 1 to the desired number of rounds):

Compute the round key: RK = K1 ⊕ (K2 <<< 3) ⊕ (K2 >>> 4).

Apply the inverse of the round function: X = F_inv(X) ⊕ RK.

Update the key halves: (K1, K2) = (K2, K1 ⊕ (K2 <<< 1) ⊕ (K2 >>> 8)).

3. Final Round:

Compute the final round key: RK_final = K1 ⊕ (K2 <<< 3) ⊕ (K2 >>> 4).

Decrypt the ciphertext: Plaintext (P) = F_inv(X) ⊕ RK_final.

Lightweight Speck

Speck aims to provide secure encryption and decryption for resource-constrained devices, such as those in the Internet of Things (IoT).Its primary goal is to strike a balance between security and efficiency, ensuring that cryptographic operations do not overwhelm limited computational resources. Speck operates on blocks of data, where each block consists of two words (e.g., 64 bits).[7][8][9]

The key size matches the block size (e.g., 128 bits for a 64-bit block). Variants of Speck allow flexibility in choosing block and key sizes. [10][11]The core of Speck's encryption process lies in its round function. Speck generates round keys from the main secret key. The key schedule involves additional rotations and XOR operations. The number of rounds and the specific key-dependent constants determine the key schedule. .[12][13]

**Encryption Steps:**

1. Initialization:

i. Input: Plaintext (P), Key (K)

ii. Divide the 128-bit key into two halves: K1 and K2.

iii. Initialize the state: X = P.

2. Rounds (Repeat for a fixed number of rounds):For each round (from 1 to the desired number of rounds):

i. Compute the round key: RK = K1 ⊕ (K2 <<< 3) ⊕ (K2 >>> 4) (where <<< denotes left rotation and >>> denotes right rotation).

ii. Apply the round function: X = F(X) ⊕ RK.

iii. Update the key halves: (K1, K2) = (K2, K1 ⊕ (K2 <<< 1) ⊕ (K2 >>> 8)).

Decryption Steps (for the same parameters):

1. Initialization:

i. Input: Ciphertext ©, Key (K)

ii. Divide the 128-bit key into two halves: K1 and K2.

iii. Initialize the state: X = C.

2. Rounds (Repeat for the same number of rounds as encryption):For each round (from 1 to the desired number of rounds):

i. Compute the round key: RK = K1 ⊕ (K2 <<< 3) ⊕ (K2 >>> 4).

ii. Apply the inverse of the round function: X = F_inv(X) ⊕ RK.

iii. Update the key halves: (K1, K2) = (K2, K1 ⊕ (K2 <<< 1) ⊕ (K2 >>> 8)).

3. Final Round

i. Compute the final round key: RK_final = K1 ⊕ (K2 <<< 3) ⊕ (K2 >>> 4).

ii. Decrypt the ciphertext: Plaintext (P) = F_inv(X) ⊕ RK_final.

## OBJECTIVES

The main challenge in the rapidly changing world of IoT is securing the communication of data while ensuring efficient system functioning, especially for resource-constrained devices. The goal of this effort is to propose a hybrid lightweight cryptographic algorithm called Salted-HybridSIMONSPECKey that enhances data privacy and integrity for IoT surveillance applications. The other major focus is to lessen the shortcomings of traditional cryptographic techniques, which are heavy on computations and cannot be utilized on low-power edge devices like the Raspberry Pi. The hybrid proposal here combines SIMON plus SPECK to minimize execution time, memory consumption, and power consumption while providing a good deal of security against any attacks to the cryptographic model.

One major research objective is to build a secure and efficient image encryption system best suited for real-time surveillance applications. The setup comprises capturing images on a Raspberry Pi-based client node, encrypting the data using the hybrid algorithm, and transmitting it over the network. A considerable breakthrough is in hash-based integrity verification; a hash value of ASCON is added to the encrypted image before the image is transmitted. This lets any tampering during the transit be detected by comparing the computed hash value with the one received. The study then evaluates the algorithm against some of the most common threats to cryptography, which include differential attacks, side-channel exploitation, and fault attacks. This adds assurance to the reliability of the proposed encryption model.

The last objective of the research is to experimentally analyze the performance in terms of efficiency, security, and robustness of the proposed hybridization. The study also takes measurements of encryption and decryption speed, memory consumption, and security strength, and compares these with the results obtained by using SIMON or SPECK alone. The avalanche effect is analyzed to quantify the sensitivity of the algorithm to input variations, establishing its resilience against brute-force and statistical attacks. The results demonstrate that the hybrid algorithm tremendously boosts performance and security; consequently, it is deemed a good candidate for IoT-based surveillance systems requiring confidentiality, integrity, and computational efficiency in real deployment environments.

## METHODS

The proposed design consists of LWC SPECK key generation algorithm SIMON algorithm considering block size of 64 bits with key size 128-bit and total number of 32 rounds of Feistel structure. Figure 1 shows the Feistel structure of SIMON round in which massage is divided into block of 64 bits , further each block is divided into 32-bit (two halves) and confusion function is applied. [10][11]. Figure 2 shows key scheduling of SPECK LWC, it involves generating round subkeys based on the original key, which are used in the encryption and decryption rounds. SPECK utilizes a Feistel network structure, where the key scheduling ensures a balanced distribution of key material across the rounds. Each round of the SPECK encryption process uses a specific set of round keys derived from the initial key through the key scheduling process. Proper design of the key scheduling algorithm contributes to the overall security of the SPECK cipher, preventing key-related vulnerabilities[14][15][16]

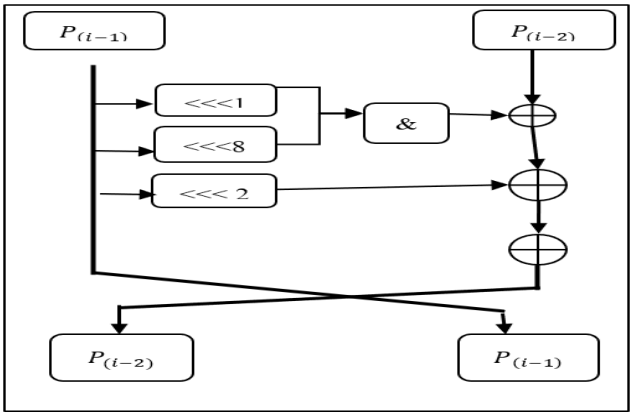


Fig 1: Round Function SIMON Algorithm

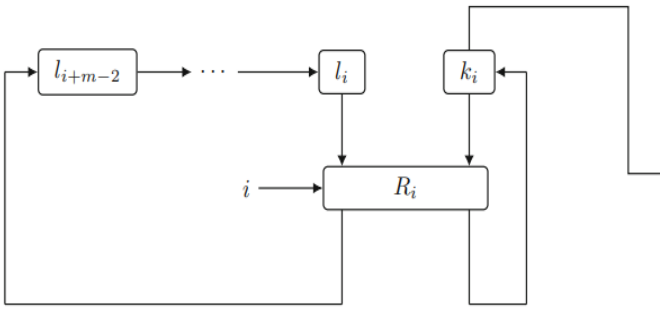


Fig 2: SPECK Algorithm Key scheduling

Implantation Of Newly Developed Hybrid Algorithm For Secure Image Transfer

Implementation of the designed and developed for secure image capturing and transferring to another node. In this web camera is attached with raspberry pi which is called as client node and transfer image to another raspberry pi (i.e sever node). The system is developed for secure transmission of data within IoT applications. Figure 3 describe the system in detail. A Client Server application is developed within the same network for demonstrating working model of the research work.

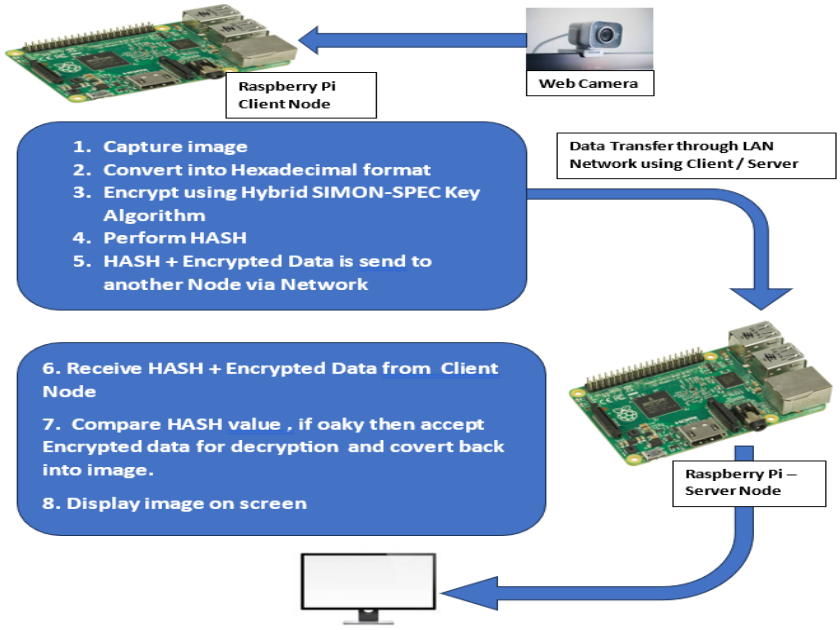


Fig 3: Working Model of Secure IoT data transfer

Step 1: Create a Server Program at first Raspbery Pi

Step 2: Create a client socket (client_socket) using the AF_INET (IPv4) and SOCK_STREAM (TCP) socket types. Capture the photo from web camera using 'fswebcam' command

Step 3: Connect to the server using the provided server_host and server_port.

Step 4: Generate a Diffie-Hellman shared secret:

a. Input the client's private key (x).

b. Calculate the client's public key (y) as y = (G^x) % P.

c. Send the client's public key (y) to the server using client_socket.

Step 5: Receive the server's public key (y_server) from the server via client_socket.

Step 6: Compute the shared secret key (shared_secret) as shared_secret = (y_server^x) % P.

Step 7: Enter an infinite loop for secure communication:

a. Input a message (message) from the client.

b. Encrypt the message using HybridSIMONSPECKey cipher with the shared_secret as the shift key

c. Generate HASH Value using ASCON-HASH Function and append with Encrypted Message

**Lightweight Hashing Used In The System To Achieve Integrity**

The Lightweight ASCON HASH is a hash function designed with a focus on efficiency and simplicity for lightweight cryptographic applications. It employs the ASCON permutation, which is a compact and lightweight design suitable for resource-constrained environments. The hashing process involves transforming input data into a fixed-size hash output.

ASCN HASH utilizes a sponge construction, where the input data is absorbed into the internal state of the algorithm in blocks. The permutation operation is then applied iteratively, ensuring that the internal state undergoes a series of transformations, creating a diffusion effect.[16][17][18]

One notable feature is its ability to adapt to various security requirements by adjusting the number of rounds in the permutation. This adaptability allows for customization based on the desired level of security and the specific constraints of the target platform. Following steps involved in ASCON Hashing

Initialization: The algorithm starts with an initial setup, initializing internal variables and parameters, including the number of rounds for the permutation.

Absorption Phase: The input message is divided into blocks, and each block is absorbed into the internal state using the ASCON permutation. This process involves bitwise operations and non-linear transformations, ensuring the input's influence on the internal state.

Padding: If necessary, padding is applied to the last block to ensure it aligns with the block size requirements of the algorithm.

Permutation Rounds: The internal state undergoes a series of permutation rounds. The number of rounds is determined by the specific configuration of the Lightweight ASCON HASH. These rounds contribute to the diffusion and mixing of the input information throughout the internal state.

Squeezing Phase:The final hash value is extracted or "squeezed" from the modified internal state. This output represents the hash of the input message.

Finalization: Any additional cleanup or finalization steps may be performed to ensure the security and integrity of the hash function.

Substitution Layer: The Substitution layer in the ASCON HASH algorithm is a crucial component responsible for introducing non-linearity and confusion within the data. It achieves this by systematically replacing or substituting elements in the internal state. This layer utilizes carefully designed substitution functions that act on different parts of the state, ensuring a complex and unpredictable transformation. The subtlety of the substitution layer lies in its ability to add a touch of chaos to the cryptographic routine, making it resistant to various attacks. It's like a carefully orchestrated chaos, where each step in the dance is purposeful, ensuring that the resulting hash is a product of intricate and unpredictable transformations [19][20][21]

## RESULTS

The Hybrid cipher's stability in outperforming SIMON in both encryption and decryption indicate that the combination of SIMON and SPECK elements contributes consistently to improved performance. The amalgamation of SIMON's rounds and SPECK's key scheduling in the hybrid algorithm results in a noteworthy enhancement in

decryption speed compared to the standalone SIMON algorithm. Decrypting a 100 KB file takes merely 4.28 seconds with the hybrid algorithm.

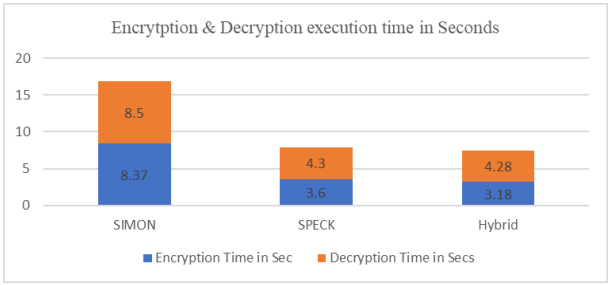


Fig 4 : Encryption & Decryption time in seconds

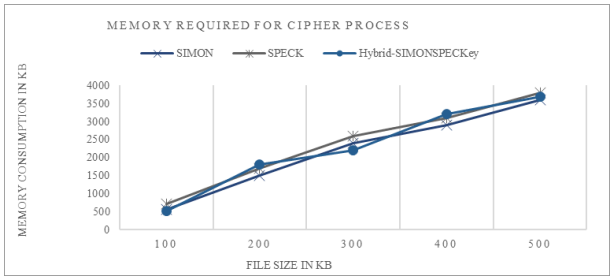


Fig 5: Memory required during Encryption (in Kbs)

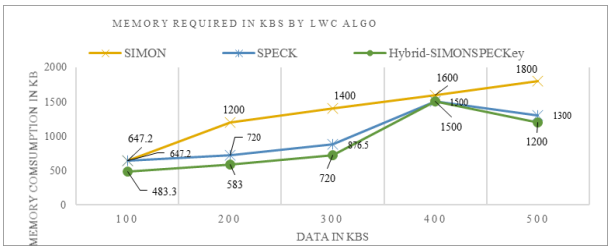


Fig 6: Memory required in Kb (decryption)

## CONCLUSION AND FUTURE WORK

The Hybrid-SIMON-SPECK Key algorithm represents a novel approach by combining the efficient round function of SIMON with the robust key scheduling mechanism of LWC SPECK.In this surveillance system, when an image is captured by the camera, it is stored on a Raspberry Pi. The Hybrid-SIMON-SPECK Key algorithm is then employed to provide confidentiality to the stored data. The encrypted data undergoes a hash function, and the resulting hash value is appended to the ciphered message. This combined data is transmitted via a LAN network to another node.

The secure transmission process involves verifying the hash value at the receiving node. If the hash value matches the computed hash from the received cipher message, it indicates the integrity of the data. Subsequently, the receiving node accepts the message and proceeds to decrypt it using the Hybrid-SIMON-SPECK Key algorithm, revealing the original image.

Therefore, through the integration of this hybrid algorithm, the Smart Surveillance System ensures both confidentiality and integrity in the data transmission and storage processes. This robust security measure enhances the overall reliability and trustworthiness of the surveillance system, making it well-suited for applications where secure image transmission and storage are paramount.

Hybrid-SIMON-SPECK algorithm shows improved result during time consumption and memory requirement and also fulfill the requirement of recourse constrained devices. As performance is improved so it become necessity to check whether security parameter of algorithm. Avalanche effect is one of the techniques to check the strength of cryptographic algorithm by calculating hamming distance. In this research work 1kb text file is passed through SIMON, SPECK & hybrid algorithm and cipher text is generated respectively. Then first character of plain text is replaced by adjacent alphabet and cipher text is generated by passing this plain text file to all algorithm respectively.

## REFERENCES

[1] Neve R, Bansode R, Kaul V. Novel Lightweight Approach to Perform Cryptography for Data Security & Privacy in IoT Mobile Devices. Int J Intell Syst Appl Eng [Internet]. 2023 Jul. 12 [cited 2023 Nov. 3];11(9s):822-8. Available from: https://ijisae.org/index.php/IJISAE/article/view/3270

[2] Muhammad Rana, Quazi Mamun, Rafiqul Islam, Lightweight cryptography in IoT networks: A survey, Future Generation Computer Systems,Volume 129,2022,Pages 77-89, https://doi.org/10.1016/j.future.2021.11.011.

[3] Michael Enriquez, Den Whilrex Garcia and Edwin Arboleda. Enhanced Hybrid Algorithm of Secure and Fast Chaos-based, AES, RSA and ElGamal Cryptosystems. Indian Journal of Science and Technology. DOI: 10.17485/ijst/2017/v10i27/105001. April 2020. Volume: 10, Issue: 27, Pages: 1-14. Available from :

https://indjst.org/articles/enhanced-hybrid-algorithm-of-secure-and-fast-chaos-based-aes-rsa-and-elgamal-cryptosystems

[4] Miguel Antonio Caraveo-Cacep, Rubén Vázquez-Medina, Antonio Hernández Zavala, A survey on low-cost development boards for applying cryptography in IoT systems, Internet of Things, Volume 22,2023,100743,ISSN 2542-6605. Available from: https://doi.org/10.1016/j.iot.2023.100743.

[5] Chaudhary RRK, Chatterjee K. (2022). A lightweight security framework for electronic healthcare system. International Journal of Information Technology. 14(6):3109-3121. https://doi.org/10.1007/s41870-022-01034-4

[6] J. Kaur and K. R. R. Kumar, "Analysis of Avalanche effect in Cryptographic Algorithms," 2022 10th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), Noida, India, 2022, pp. 1-4, Available from : doi: 10.1109/ICRITO56286.2022.9965127.

[7] Rahul Neve, Rajesh Bansode, "Review of Lightweight Cryptography for Secure Data Transmission in Resource Constraint Environment of IoT" , Book chapter Computing and Communications Engineering in Real-Time Application Development , 1st edition ,22 September 2022,ISBN 9781003277217, Available from : https://doi.org/10.1201/9781003277217

[8] Lo'ai, Tawalbeh, Michael Alicea, Izzat Alsmadi,New and Efficient Lightweight Cryptography Algorithm for Mobile and Web Applications,Procedia Computer Science,Volume 203,2022,Pages 111-118,ISSN 1877-0509, Available from :https://doi.org/10.1016/j.procs.2022.07.016

[9] D. Upadhyay, N. Gaikwad, M. Zaman and S. Sampalli, "Investigating the Avalanche Effect of Various Cryptographically Secure Hash Functions and Hash-Based Applications," in IEEE Access, vol. 10, pp. 112472-112486, 2022, Available from: https:// doi: 10.1109/ACCESS.2022.3215778.

[10] Fatma K Tabash, M. Izharuddin, Mosab I Tabash, Encryption techniques for H.264/AVC videos: A literature review, Journal of Information Security and Applications, Volume 45,2019,Pages 20-34,ISSN 2214-2126, Available from : https://doi.org/10.1016/j.jisa.2019.01.001.

[11] Nada Alhirabi, Omer Rana, and Charith Perera. 2021. Security and Privacy Requirements for the Internet of Things: A Survey. ACM Trans. Internet Things 2, 1, Article 6 (February 2021), 37 pages. Available from : https://doi.org/10.1145/3437537

[12] Sunny Sall1, Rajesh Bansode, " Lightweight Cryptography Using Pairwise Key Generation and Malicious Node Detection in Large Wireless Sensor Network" , Indian Journal of Science and Technology , Year: 2023, Volume: 16, Issue: 36, Pages: 3002-3008 , Available from : doi:10.17485/IJST/v16i36.2503

[13] S. D. Sanap and V. More, "Performance Analysis of Encryption Techniques Based on Avalanche effect and Strict Avalanche Criterion," 2021 3rd International Conference on Signal Processing and Communication (ICPSC), Coimbatore, India, 2021, pp. 676-679, doi: 10.1109/ICSPC51351.2021.9451784.

[14] N. M. Karie, N. M. Sahri, W. Yang, C. Valli and V. R. Kebande, "A Review of Security Standards and Frameworks for IoT-Based Smart Environments," in IEEE Access, vol. 9, pp. 121975-121995, 2021, doi: 10.1109/ACCESS.2021.3109886.

[15] V. Hassija, V. Chamola, V. Saxena, D. Jain, P. Goyal and B. Sikdar, "A Survey on IoT Security: Application Areas, Security Threats, and Solution Architectures," in IEEE Access, vol. 7, pp. 82721-82743, 2019, doi: 10.1109/ACCESS.2019.2924045.

[16] F. T. Jaigirdar, B. Tan, C. Rudolph and C. Bain, "Security-Aware Provenance for Transparency in IoT Data Propagation," in IEEE Access, vol. 11, pp. 55677-55691, 2023, doi: 10.1109/ACCESS.2023.3280928.

[17] Caudhari, A., Bansode, R. (2021). Securing IoT Devices Generated Data Using Homomorphic Encryption. In: Balas, V.E., Semwal, V.B., Khandare, A., Patil, M. (eds) Intelligent Computing and Networking. Lecture Notes in Networks and Systems, vol 146. Springer, Singapore. https://doi.org/10.1007/978-981-15-7421-4_20

[18] Zitouni, N., Sedrati, M. & Behaz, A. LightWeight energy-efficient Block Cipher based on DNA cryptography to secure data in internet of medical things devices. Int. j. inf. tecnol. (2023). https://doi.org/10.1007/s41870-023-01580-5

[19] Li, H., Yang, G., Ming, J. et al. Transparency order versus confusion coefficient: a case study of NIST lightweight cryptography S-Boxes. Cybersecur 4, 35 (2021). https://doi.org/10.1186/s42400-021-00099-1

[20] Cherbal, S., Zier, A., Hebal, S. et al. Security in internet of things: a review on approaches based on blockchain, machine learning, cryptography, and quantum computing. J Supercomput (2023). https://doi.org/10.1007/s11227-023-05616-2

[21] Neve, R.P., Bansode, R. (2023). Performance Evaluation of Lightweight ASCON-HASH Algorithm for IoT Devices. In: Balas, V.E., Semwal, V.B., Khandare, A. (eds) Intelligent Computing and Networking. IC-ICN 2023. Lecture Notes in Networks and Systems, vol 699. Springer, Singapore. https://doi.org/10.1007/978-981-99-3177-4_25