JCST Journal of Data Acquisition and Processing

ISSN: 1004-9037
https://sjcjycl.cn/
DOI: 10.5281/zenodo.7778508

ANALYSIS AND DEVELOPMENT OF SECURITY FRAMEWORK FOR IOT DEVICE

Abhinandan Singh Dandotiya[1], Dr. Shashikant Gupta[2]
1,Department of Computer Science Engineering,ITM University, Gwalior
2.,Department of Computer Science Engineering,ITM University, Gwalior
Abhinandan.dandotiya@gmail.com, shashikantgupta@itmuniversity.ac.in

Abstract:

Internet of Things (IoT) devices are growing rapidly, making security and privacy crucial. This study analyses and develops an IoT-specific security framework. To address the specific security problems of IoT devices and provide effective techniques and measures to safeguard sensitive data, minimize vulnerabilities, and assure IoT system integrity. The analysis phase identifies and evaluates IoT ecosystem security issues such poor authentication, data encryption, susceptible firmware and software, and lack of standardization. The framework examines these difficulties to understand the security landscape and build effective countermeasures. The analysis informs the development phase, which includes device authentication and access control, data encryption and privacy protection, secure firmware and software upgrades, and standardization and compliance. These steps increase IoT device security by guaranteeing secure communication, data integrity, and protection against unauthorized access and assaults. This Paper Proposed security framework, to assure the security framework's originality and efficacy, it is developed methodically. IoT security literature, best practices, and upcoming technologies are researched extensively. The suggested security architecture is projected to improve IoT device trustworthiness and reliability, promoting their wider usage across domains and generating lightweight cryptography techniques. This analysis-driven and comprehensive methodology addresses security issues to help build secure and resilient IoT ecosystems.

Keywords: Internet of Things (IoT), security framework, analysis, development, authentication, data encryption,

1. Introduction:

The Internet of Things (IoT) has emerged as a revolutionary force in our increasingly interconnected society, effortlessly integrating smart devices into our daily lives. IoT technology has transformed the way we live, work, and interact with our surroundings, from smart homes and wearable gadgets to industrial systems and healthcare applications. However, as the number of IoT devices grows, so does the need to address the inherent security risks that come with their widespread adoption. This article provides an in-depth examination and development of a comprehensive security framework designed exclusively for IoT devices. This framework intends to provide effective strategies and ways to enhance the security of these devices, secure sensitive data, and assure the continuous operation of IoT systems by examining the weaknesses and hazards existing in IoT ecosystems. The introduction contextualizes the growing popularity of IoT devices and emphasizes the importance of addressing the security challenges connected with their proliferation. Following that, the article will go into the examination of IoT security concerns and the subsequent construction of a security architecture. We investigate the multidimensional nature of IoT security in this study, taking into account variables such as data integrity, confidentiality, device authentication, and network resilience. Recognizing the particular vulnerabilities provided by IoT devices, we aim to create a framework that mitigates risks and instills trust in consumers, supporting the long-term growth and adoption of IoT technology[1][2][3][4]. The conclusion summarizes the introduction, emphasising the significance of a complete security framework for IoT devices and laying the groundwork for additional research into the analysis and implementation of effective security solutions.The rest of our research is structured as follows.A literature review in Section 2. In section 3, we propose a security framework and lightweight cryptography technique that improves the security of Internet of Things devices. In Section 4, describe the security analysis that the security framework accomplishes. Section 5 concludes with findings and suggestions for future research.

2. Literature Survey –

Security of heterogeneous network devices is one of the security challenges. Traditional security solutions proposed and developed over the years have been rendered ineffective and infeasible for IoT applications due to the unique nature of these devices. However, various lightweight solutions for IoT applications have been offered, although they are far from efficient. IoT device makers confront energy and data security issues. Even with application layer security updates, these risks and issues are becoming more common, especially when low-resource devices transfer sensitive data [5,6].These are some of the latest IoT security frameworks available in the literature. Researchers and industry professionals continue to develop and refine frameworks to address the evolving IoT security landscape. It is advisable to explore these frameworks further by referring to the respective papers for a more comprehensive understanding of their concepts and methodologies. SECoS focuses on securing IoT communications by proposing a lightweight and efficient framework. It incorporates cryptographic techniques, secure routing protocols, and efficient key management mechanisms to enhance the security of IoT networks. SHIELD handles heterogeneous IoT security issues. It secures IoT devices via trust management, device integration, and data transmission. SIFA offers measuring IoT device security and functionality. To assess and improve IoT security, it uses risk-based testing, vulnerability analysis, and security. iCoreSec integrates device authentication, secure communication, access management, and anomaly detection into an IoT security framework. It offers complete IoT security solutions. MAMID authenticates IoT devices. It proposes using machine-to-machine (M2M) authentication protocols to build trust between IoT devices for secure and authorised communication. PRoSPECT secures resource-constrained IoT devices. It blends threat modelling, secure coding, and vulnerability assessment into IoT system development with a process-based security architecture. The authors examine major IoT frameworks as Contiki, TinyOS, OpenWSN, IoTivity, AllJoyn, and OSGi. They assess the security features provided by these frameworks, highlighting their strengths and limitations.The framework aims to address the security challenges faced by IoT systems, considering the unique characteristics and requirements of IoT environments.[7][8][9][10][11][12]

Maitra and Paul (2008) used the KSA phase with zigzag and IV replacement to RC4+ for safety. PRGA shift operation pointers.[13] The RC4-2S algorithm with S-box split by Hammood et al. (2013) increased key stream randomization[14]. Jindal and Singh (2017) updated three RC4 algorithms using RC4+ to reduce encryption time and boost key stream randomness[15]. Weerasinghe (2012) improved RC4 algorithm secrecy [16]. Authors constructed successful double S-box RC4 utilizing modified proposed approach.

Kang et al. (2021) proposed a two-tier privacy-preserving data inference approach to reduce transmission data and battery usage from sensed data. Authors protected sensitive data from enemies[17]. Xu (2020) provided light-weight secure IoT (LS-IoT) with lightweight access control for real-time physical activity analysis for the physio net challenges database[18]. Ullah et al. (2021) examined Fog computing designs for safe transmission and data collection[19]. Taxonomy classifies schemes. Durairaj and Muthuramalingam (2019) explored IoT data encryption using AES-RSA-ECC[20]. Communication, devices, cloud, and main are IoT levels. IoT application, connections, gateway, cloud, devices, and users create these levels. Multistage encryption protects cloud layer. AES-encrypted cloud messaging. ECC's private key decrypts the message encrypted by RSA's public and symmetric keys. Chandu et al. (2017) proposed hybrid cloud IoT data encryption and security[21]. AES receives cloud-encrypted data. Authorized users send RSA-encrypted AES keys. Nikravan and Reza (2020) used IoT MFA[22]. The proposed protocol comprises three stages: Session Key, Mutual authentication of IoT devices and users, and Multifactor authentication.

Buffer overflows, viruses, Trojans, and worms are vulnerable. AES and RSA reduce these dangers. Huang et al. (2017) revised IoT access control fog computing and Cypher text outsourcing. before cloud storage. Authorized users can decrypt data[23]. Simulations show successful computational activity and 2% longer encryption and decoding.Petrvalsky and Drutarovsky (2016) proposed a microcontroller-friendly differential power analysis (PDA) assault countermeasure[24]. Secure embedded devices randomly assign intermediate value general constant weight codes. Data hamming weight for each value balances power utilization and complicates DPA attack. Table-based AES encryption decreases demonstration table size. Aerabi et al. (2020) MCU-based ultra-low-energy IoT devices use secure communication. The design assesses, compares security, and finds energy-consuming COTS in the IoT system[25].Heterogeneous network devices present distinct issues in IoT security. Resource limits and data sensitivity make traditional security solutions unsuitable for IoT applications. Researchers and industry people have suggested IoT security frameworks that use cryptography to reduce these concerns. SECoS, SHIELD, SIFA, iCoreSec, MAMID, and PRoSPECT are lightweight and efficient frameworks for IoT communications, device integration, data transmission, and functionality evaluation. To prevent buffer overflows, unauthorized access, and data breaches, these frameworks use cryptographic algorithms including RC4, AES, RSA, ECC. Microcontroller-friendly differential power analysis and cloud-based encryption techniques provide safe and energy-efficient IoT device connectivity[26][27][28][29][30]. These frameworks and cryptographic methods are improving IoT security by protecting sensitive data and improving system security.

Framework	Description	Cryptographic Techniques
SECoS	Securing IoT communications with cryptographic techniques, secure routing, and key management.	RC4, AES, RSA, ECC
SHIELD	Addressing security challenges in heterogeneous IoT environments with trust management and secure data transmission.	AES, RSA, ECC
iCoreSec	Comprehensive IoT security framework including device authentication, secure communication, access control, and anomaly detection.	RC4, AES, RSA, ECC
MAMID	Focusing on IoT device authentication with machine-to-machine (M2M) authentication protocols.	RC4
PRoSPECT	Addressing security challenges in resource-constrained IoT devices through the development lifecycle.	AES, RSA, ECC

Table 1: Framework Analysis

Secure and strong frameworks are essential in the ever-changing digital ecosystem.in which Lightweight timing-based cryptography improves resource-constrained device performance. Data privacy and protection are crucial in IoT and cloud computing. As resource-constrained devices grow, lightweight cryptography, which reduces computing overhead and delay, is crucial. Lightweight encryption techniques that account for timing issues can provide solid security with low-resource device computational and temporal overhead. Modern, secure frameworks need these lightweight cryptographic solutions to protect sensitive data, ensure confidentiality and integrity, and mitigate timing-based attacks. Lightweight cryptography research must integrate computational efficiency and speed to suit digital security concerns and modern computing paradigms.

3. Proposed Methodology

Sensitive IoT systems need security and privacy. Using device authentication, data encryption, access control, intrusion detection, physical security, and incident response, we can secure IoT systems.

More families and businesses are linking common objects to the internet with IoT devices. IoT devices can be attacked, therefore widespread use poses security concerns. IoT implementations need security assessments. This the research work proposed IoT security framework:

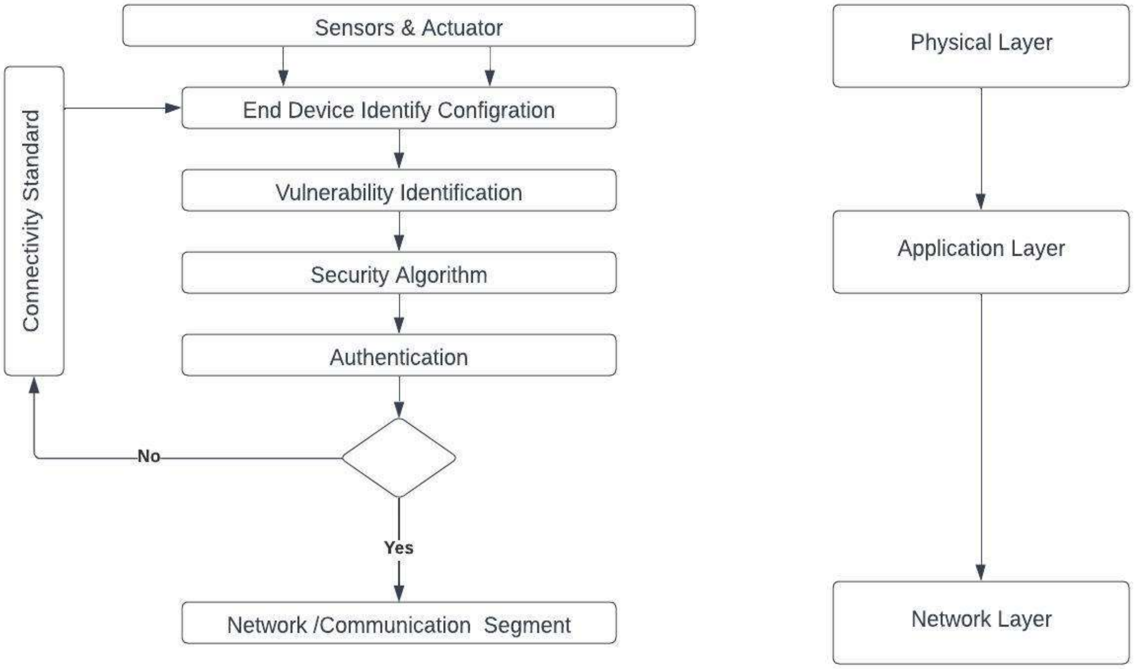


Figure 1: Proposed Framework

This framework first Evaluating IoT system configuration. Testing system configuration, operation, and connectivity. In Second Step IoT vulnerability detection and protection begin with identification. Assessing risks. Checklists evaluate security. In next step Encrypted sensor/actuator data. Data transfer equipment tamper-proofing. Receives encrypted data. The security assessment framework transfers sensor-standard data between components, with the end component managing vulnerability and validating the data delivery mechanism. Connectivity standards assess system integrity before returning data to the starting component if the series of components fails. It shields the system. After meeting criteria, data is sent to the next tier.To achieving the goal of IoT Security the Lightweight cryptography mechanism is encompasses with this proposed framework.This paper proposes a lightweight cryptography technique (LWCT)to authenticating data and guaranteeing privacy.

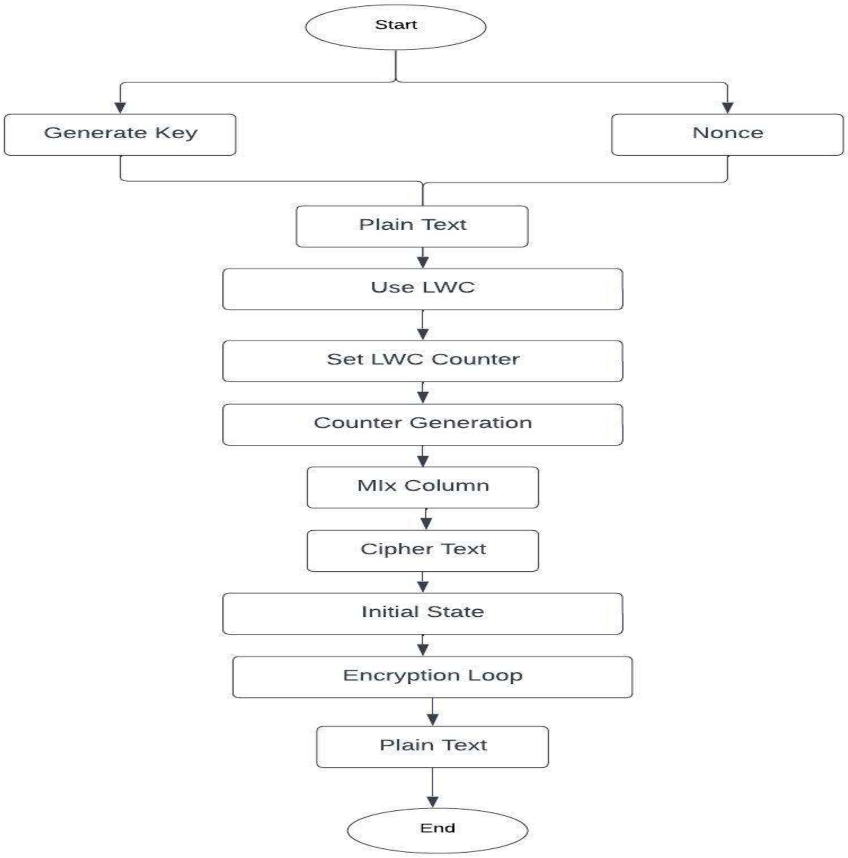


Figure 2: Proposed Flowchart of LWCT

As IoT technology advances, data transmission over the network must be safe. Traditional network access control methods are easily cracked or replicated. Ciphertext Attackers can easily obtain encryption keys and restore plaintext. A LWCT method protects IoT and device data. With the Key generation (KG) and Random Number Generation .

The suggested technique processes Lightweighted Cipher(LWC). It improves security and efficiency. The Lightweighted Cryptography Technique(LWCT) encrypts plaintext to cipher text.Modern algorithms, protocols, and systems use this encryption.

– KeyGenerate (KG) Generate the Key

- The 256-bit encryption key must be kept secret.

- Key Expansion: The secret key (Puk) expands the key into 32-bit words and constants used in encryption.

- The nonce is a random value that must be unique for each encryption process but need not be kept secret.

Algorithm 1 shows the proposed LWTC Encryption Process.

Algorithm 1

Input parameters: input data parameters

Output: Cipher Text

1. Set length as a random byte length integer.
2. set random_byte as an empty sequence

```
3. Add a random byte b from 0 to 255 to the random_bytes sequence.
4. KG: Private key PuK received randome_byte.
5. Pad the key with zeroes if PuK is less than 32 bytes.
6. Do Key Expansion here.
7. Generate 64-bit random nonce (IV).
8. Use LWC as the core primitive.
    a. Set LWC constant and initial state using expanded key and nonce.
    b. For 64-bit plaintext.
    c. Counter generation: increase each block's 32-bit counter for uniqueness.
    d. Mix Column: Apply LWC quarter round function to state, mixing column to
       block.

       The LWC quarter-round function diffuses values by mixing four 32-bit words
       (a, b, c, d). Steps include:
       Add : a = a+b , d=d+c
       XOR : d=d xor a , b = b xor d
       Rotate : a= (a<<<16) , c= (c <<< 12)
       ADD : a=a+b , d=d+c
       XOR : d= d XOR a , b= b XOR d
       Rotate : a= (a <<< 8) , c=(c<<<7)
       (<<< denote the left rotation)
9. Encrypted 64-bit ciphertext is obtained after processing all blocks.
```

Algorithm 2 shows the proposed LWTC Decryption Process.

Algorithm 2

```
Input : Cipher text, Key, nonce
Output: Plain Text
1. Initialization: Constants
       sigma = 512 bit constant 'expand 32 byte k'
       tau = 512 bit constant 'nonce constant'
2. Initial State (16-32 bit word 'm')
   m[0] to m[3] : the constant sigma[0] to sigma [3]
   m[4] to m[11] : the 128 bit key (divided into 8 32 bit words)
   m[12] : the block cipher
   m[13] to m[15] : the 128 bit block nonce
3. Encryption Loop:
   • The State m is copied into a working array a.
   • LWC quarter rounds (20 round. 10 iterations):
       o For each quarter round the asking array a undergoes the following
         transformation –
       o a[0] = a[0]+ a[4] ; a[12] = (a[12] ^ a[0]) <<<16
       o a[8] =a[8]+a[12] ; a[4] =(a[4] ^a[8]) <<<12
       o a[0] = a[0]+a[4] ; a[12] = (a[12] ^ a[0]) <<<8
       o a[8] = a[8] +a[12] ; a[4] = (a[4] ^ a[8]) <<<7
```

4. Update state after 20 cycles by adding working array "a" to original state "a" (mod 2^32).
5. The key stream is obtained by adding (mod 2 ^32) updated state "m" to beginning state.\
6. The key stream is XORed with the ciphertext blocks to decrypt the plaintext.

4. Result and Discussion-

4.1 Experiment Analysis

Encryption, decryption, secrecy, and throughput are used to evaluate the suggested technique. Table 2 lists the model's system features and parameters.

System	**CPU**	**Intel i5 (3.2 GHZ)**
Configuration	Python	Cryptography Classes
Model	Key-Size (bits)	Block-Size
	RAM	8 GB
	Operating System	Windows 8
	System Types	64 Bits

Table 2: Simulation Setting

4.2 Performance Analysis –

4.2.1 Analysis of Encryption Time –

Table 3 compares the proposed cryptography technique LWTC to various current algorithms for encryption time, The suggested approach achieves an 0.06 **µs**for all data and provides quick encryption. The encryption time analysis is depicted in Figure 3.

Comparative analysis of encryption time

Reference	**Algorithm**	**PlainText Size**	**CipherText Size**	**Key Size**	**Encryption Time (µs)**
31	Skinny-64-128	64 bits	128 bits	128 bits	0.13
32	TWINE-128	64 bits	64 bits	128 bits	0.14
33	Sparx-128-128	128 bits	128 bits	128 bits	0.16
34	Hummingbird-2 (HBC)	256 bits	256 bits	128 bits	0.18
35	ISAP (128A, 128B)	64 bits	64 bits	128 bits	0.11
36	TEA (Tiny Encryption Algorithm)	64 bits	64 bits	128 bits	0.15
37	XTEA (Extended Tiny Encryption Algorithm)	64 bits	64 bits	128 bits	0.12
38	LED	64 bits	64 bits	128 bits	0.16
39	HIGHT-Cipher	64 bits	64 bits	128 bits	0.11
40	PRESENT-128	64 bits	64 bits	128 bits	0.13

	Proposed LWCT	**64 bits**	**64 bits**	**256 bits**	0.06

Table 3: Comparative analysis of encryption time

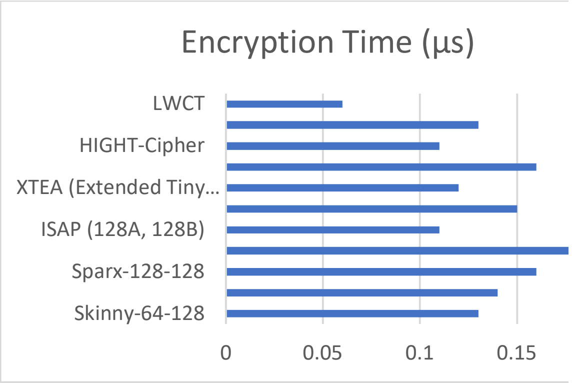
4.2.2 Analysis of Decryption Time –

Table 4 compares the decryption time analysis of the proposed technique to that of existing approaches. In comparison to previous techniques, the suggested model appears to have a 0.05 s shorter decryption time. Figure 4 depicts the decryption time graphs for the proposed and existing methods.

Comparative analysis of decryption time –

Algorithm	**PlainText Size**	**CipherText Size**	**Key Size**	**Decryption Time (µs)**
Skinny-64-128	64 bits	128 bits	128 bits	0.12
TWINE-128	64 bits	64 bits	128 bits	0.14
Sparx-128-128	128 bits	128 bits	128 bits	0.15
Hummingbird-2 (HBC)	256 bits	256 bits	128 bits	0.18
ISAP (128A, 128B)	64 bits	64 bits	128 bits	0.11
TEA (Tiny Encryption Algorithm)	64 bits	64 bits	128 bits	0.15
XTEA (Extended Tiny Encryption Algorithm)	64 bits	64 bits	128 bits	0.12
LED	64 bits	64 bits	128 bits	0.16
HIGHT-Cipher	64 bits	64 bits	128 bits	0.11
PRESENT-128	64 bits	64 bits	128 bits	0.13
Proposed LWCT	**64 bits**	**64 bits**	**256 bits**	**0.05**

Table 4: Comparative analysis of decryption time



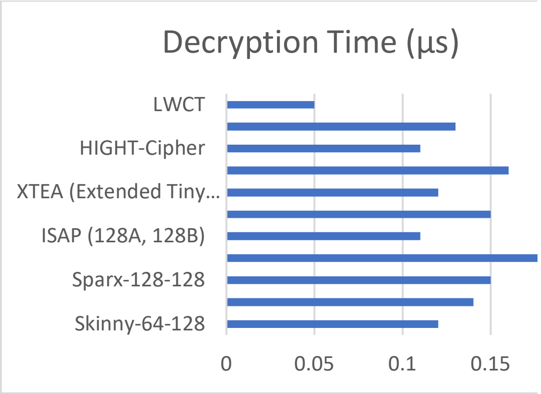


Figure 3 : Encryption Time Analysis

Figure 4: Decryption Time Analysis

4.2.3 Analysis of Encryption Throughput –

Data/encryption time is used to calculate encryption throughput. As a result, when encryption throughput increases, the efficiency of the approaches is considered. Table 5 shows the analytical data and encryption throughput of 2 Gbps. Eq. 1 is used to compute encryption throughput.

Figure 5 depicts a study of encryption throughput in comparison to other approaches.

Encryption thorughput (bits/**µ**s) =Σ(input data) / Σ(encryption time) ---------------(1)

Algorithm	**PlainText Size**	**CipherText Size**	**Key Size**	**Encryption Throughput (Gbps)**
Skinny-64-128	64 bits	128 bits	128 bits	7.38
TWINE-128	64 bits	64 bits	128 bits	4.57
Sparx-128-128	128 bits	128 bits	128 bits	6.4
Hummingbird-2 (HBC)	256 bits	256 bits	128 bits	11.56
ISAP (128A, 128B)	64 bits	64 bits	128 bits	4.57
TEA (Tiny Encryption Algorithm)	64 bits	64 bits	128 bits	3.73
XTEA (Extended Tiny Encryption Algorithm)	64 bits	64 bits	128 bits	4.67
LED	64 bits	64 bits	128 bits	3.5
HIGHT-Cipher	64 bits	64 bits	128 bits	5.09
PRESENT-128	64 bits	64 bits	128 bits	4

Proposed LWCT	**64 bits**	**64 bits**	**256 bits**	**2**

Table 5: Comparative analysis of encryption throughput

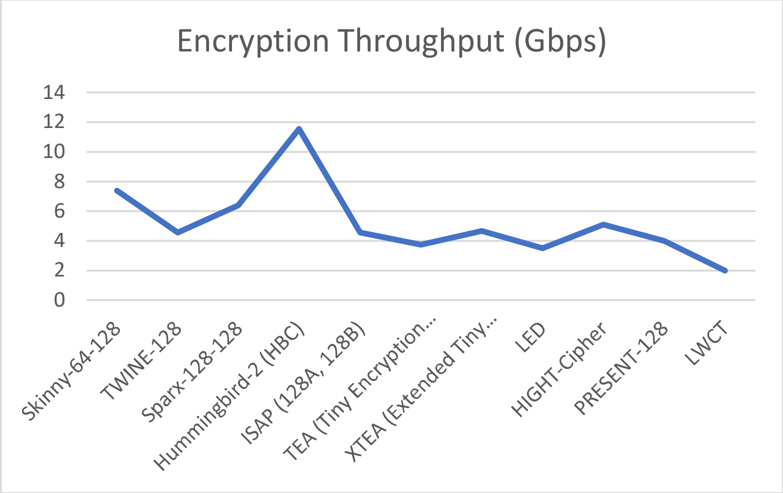


Figure 5 :Encryption Throughput Time Analysis

4.2.4 Analysis of Decryption Throughput –

Time to decrypt the input file. Table 6 shows that the decryption throughput for a data is 2 Gbps.Eq. 2 defines the decryption throughput computation. The decryption throughput analysis is depicted in Figure 6.Decryption thorughput (bits/**μ**s) =Σ(cipher text) / Σ(decryption time) ---------------(2)

Algorithm	**PlainText Size**	**CipherText Size**	**Key Size**	**Decryption Throughput (Gbps)**
Skinny-64-128	64 bits	128 bits	128 bits	8
TWINE-128	64 bits	64 bits	128 bits	4.57
Sparx-128-128	128 bits	128 bits	128 bits	6.93
Hummingbird-2 (HBC)	256 bits	256 bits	128 bits	11.56
ISAP (128A, 128B)	64 bits	64 bits	128 bits	4.57
TEA (Tiny Encryption Algorithm)	64 bits	64 bits	128 bits	3.73
XTEA (Extended Tiny Encryption Algorithm)	64 bits	64 bits	128 bits	4.67
LED	64 bits	64 bits	128 bits	3.5
HIGHT-Cipher	64 bits	64 bits	128 bits	5.09
PRESENT-128	64 bits	64 bits	128 bits	4
Proposed LWCT	**64 bits**	**64 bits**	**256 bits**	**2**

Table 6: Comparative analysis of decryption throughput

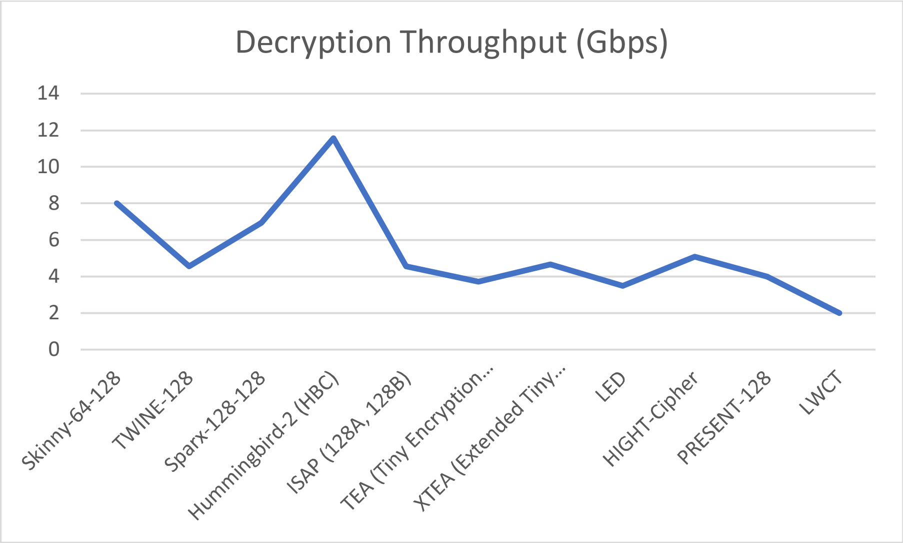


Figure 6:Encryption Throughput Time Analysis

4.2.5 Security Analysis

The secrecy level of the suggested model is studied and compared to the results of existing models. Figure 9 displays a comparison of each model's level of secrecy using the proposed method. Intruders will struggle to persuade the network to access inbound information or data from IoT devices due to the high security level. In terms of security needs such as secrecy, authentication, assaults, integrity, and confidentiality, Table 7 compares the proposed approach to various existing alternatives. The proposed LWCT models show that they meet all security standards.

Algorithm	Key Agreement	Integrity	Confidentiality	Secrecy	Resident to Man-in-the-Middle Attack	Resident to Malicious User Attack	Resident to Insider Attack	Resident to Brute Force Attack	Key Exchange
Skinny-64-128	X	✓	X	X	✓	X	X	X	X
TWINE-128	✓	✓	✓	X	✓	X	X	X	X
Sparx-128-128	X	X	X	X	X	X	X	X	X
Hummingbird-2 (HBC)	✓	✓	✓	✓	✓	✓	✓	✓	✓
ISAP (128A, 128B)	X	✓	X	X	X	✓	X	X	✓

TEA (Tiny Encryption Algorithm)	✓	✓	✓	X	✓	X	X	X	✓
XTEA (Extended Tiny Encryption Algorithm)	✓	✓	✓	X	✓	X	X	X	X
LED	✓	✓	✓	X	✓	X	X	X	X
HIGHT-Cipher	✓	✓	✓	✓	✓	✓	✓	✓	✓
PRESENT-128	✓	✓	✓	✓	✓	✓	✓	✓	✓
Proposed LWCT	✓	✓	✓	✓	✓	✓	✓	✓	✓

Table 7 compares the proposed approach to various existing alternatives

5. Conclusion

The recommended algorithm LWCT is used to consider data security for IoT-based systems. This integrated and recommended solution improves data security from IoT devices to facilities and research institutions. The method and key generation mechanism improve the key encryption and decryption procedure, which aids in preventing unauthorized people from accessing the data. The proposed approach provides very low encryption and decryption times, exceeding other current solutions.

6. **Future Work**

Future study will consider massive amounts of health data at real-time transmission with longer encryption and decryption times.

7. Acknowledgements

We would like to express our heartfelt gratitude to everyone who helped this research. In recognizing their valuable efforts.

8. References-

[1]. D. E. Kouicem and A. Mehaoua, “Security challenges in the Internet of Things: A comprehensive study,” *J. Netw. Comput. Appl.*, vol. 84, pp. 38-54, 2017. doi:10.1016/j.jnca.2016.12.006.

[2]. J. Granjal et al., “Security for the Internet of Things: A survey of existing protocols and open research issues,” *IEEE Commun. Surv. Tutor.*, vol. 17, no. 3, pp. 1294-1312, 2015. doi:10.1109/COMST.2015.2413003.

[3]. X. Fu et al., “Security and privacy in the Internet of Things: A survey,” *IEEE Internet Things J.*, vol. 4, no. 5, pp. 1250-1258, 2017. doi:10.1109/JIOT.2017.2768290.

[4]. G. Karabulut Kurt and A. Albayrak, “Security issues and challenges in IoT systems: A comprehensive survey,” *J. Netw. Comput. Appl.*, vol. 173, p. 102862, 2021. doi:10.1016/j.jnca.2021.102862.

[5]. Aljazeera.K.R, et al, “Design and characterization of BlockCryptocore.”, International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES), 2017.

[6]. M. Al-Shatari et al., “An efficient implementation of LED block cipher on FPGA”, International Conference of Intelligent Computing and Engineering (ICOICE), 2019. doi:10.1109/ICOICE48418.2019.9035193.

[7]. M. Ammar et al., “Internet of Things: A survey on the security of IoT frameworks,” *J. Inf. Sec. Appl.*, vol. 38, pp. 8-27, 2018. doi:10.1016/j.jisa.2017.11.002.

[8]. A. Ali et al., “Advanced security framework for Internet of things (IoT),” *Technologies*, vol. 10, no. 3, p. 60, 2022. doi:10.3390/technologies10030060.

[9]. A. Ali et al., “Advanced security framework for Internet of things (IoT),” *Technologies*, vol. 10, no. 3, p. 60, 2022. doi:10.3390/technologies10030060.

[10]. J. Pacheco et al., “IoT security framework for smart water system” in 14th International Conference on Computer Systems and Applications (AICCSA), vol. 2017. IEEE/ACS. IEEE, 2017, Oct., pp. 1285-1292. doi:10.1109/AICCSA.2017.85.

[11]. K. Balasamy et al., “A secure framework for protecting clinical data in medical IoT environment,” *Smart Healthc. Syst. Des. Sec. Privacy Aspects*, pp. 203-234, 2022.

[12]. R. R. K. Chaudhary and K. Chatterjee, “A lightweight security framework for electronic healthcare system,” *Int. J. Inf. Technol.*, vol. 14, no. 6, pp. 3109-3121, 2022. doi:10.1007/s41870-022-01034-4.

[13]. S. Maitra and G. Paul, “Analysis of RC4 and proposal of additional layers for better security margin,” *Prog. Cryptol.-INDOCRYPT*, *Proc. 9*: 9th International Conference on Cryptology in India, Kharagpur, India, December 14-17, 2008. Berlin Heidelberg: Springer, 2008.

[14]. M. M. Hammood et al., *RC4-2S: RC4 Stream Cipher with Two State Tables*. Security: Information Technology Convergence*, Robotics, Automations and Communication*. Dordrecht: Springer Netherlands, 2013, pp. 13-20.

[15]. S. S. Dhanda et al., “Lightweight cryptography: A solution to secure IoT,” *Wirel. Personal Commun.*, vol. 112, no. 3, pp. 1947-1980, 2020. doi:10.1007/s11277-020-07134-3.

[16]. T. D. B. Weerasinghe, “Analysis of a modified RC4 algorithm,” *Cryptol. Eprint Arch.*, 2014.

[17]. J. J. Kang, et al., “An energy-efficient and secure data inference framework for internet of health things: A pilot study,” *Sensors (Basel)*, vol. 21, no. 1, p. 312, 2021. doi:10.3390/s21010312.

[18]. P. Rana and B. P. Patil, “Cyber security threats in IoT: A review,”,*JHS*, vol. 29, no. 2, pp. 105-120. doi:10.3233/JHS-222042.

[19]. F. Ullah, et al., “Risk management in sustainable smart cities governance: A TOE framework,” *Technol. Forecasting Soc. Change*, vol. 167, p. 120743, 2021. doi:10.1016/j.techfore.2021.120743.

[20]. K. Dewangan et al., “A review: A new authentication protocol for real-time healthcare monitoring system,” *Ir. J. Med. Sci.*, vol. 190, no. 3, pp. 927-932, 2021. doi:10.1007/s11845-020-02425-x.

[21]. S. S. Kumar and M. S. Koti, “RETRACTED ARTICLE: An hybrid security framework using internet of things for healthcare system” *Netw. Model. Anal. Health Inform. Bioinformatics*, vol. 10, no. 1, p. 52, 2021. doi:10.1007/s13721-021-00329-z.

[21]. S. S. Kumar and M. S. Koti, “RETRACTED ARTICLE: An hybrid security framework using internet of things for healthcare system” *Netw. Model. Anal. Health Inform. Bioinformatics*, vol. 10, no. 1, p. 52, 2021. doi:10.1007/s13721-021-00329-z.

[22]. G. Huang, et al., *‘Multi-scale dense convolutional networks for efficient prediction.’ arXiv Preprint ArXiv:1703.09844 2.2*, 2017.

[23]. Petrvalsky et al., “Compact FPGA hardware platform for power analysis attacks on cryptographic algorithms implementations,” *ActaElectrotechn. Inform.*, vol. 16, no. 2, pp. 3-7, 2016.

[24]. “Al_Azzawi, RuahMouadAlyas, and SUFYAN SALIM AL-DABBAGH,” Software Implementation Solutions of A Lightweight Block Cipher to Secure Restricted IoT Environment: A Review *AL-Rafidain Journal of Computer Sciences and Mathematics* 16.2 (2022), pp. 77-88.

[25]. J. Anderson et al., “Efficient and secure use of cryptography for watermarked signal authentication,”, *International Technical Meeting of the The Institute of Navigation*, *Proc. 2022 International Technical Meeting of the Institute of Navigation*, 68-82, 2022. doi:10.33012/2022.18228.

[26]. A. Khurshid, et al., “ShieLD: Shielding cross-zone communication within limited-resourced IoT devices running vulnerable software stack,” *IEEE Trans. Depend. Sec. Comput.*, vol. 20, no. 2, pp. 1031-1047, 2022. doi:10.1109/TDSC.2022.3147262.

[27]. S. Saha, Revisiting fault analysis of block ciphers: Attacks, defenses, and vulnerability assessment frameworks [Diss.] IIT Kharagpur, 2021.

[28]. A. Botello, et al, “An agent architecture for large-scale security simulation,” Tech. rep. Information Sciences Institute. Marina del Rey, CA, 2011.

[29]. R. F. Brukhanskyi and I. V. Spilnyk, “Crypto assets in the system of accounting and reporting,” *Probl. Econ.*, vol. 2, no. 40, pp. 145-156, 2019. doi:10.32983/2222-0712-2019-2-145-156.

[30]. S. Sun, et al., “Analysis of AES, SKINNY, and others with constraint programming,” *IACR Trans. Symmetric Cryptol.*, vol. 2017, no. 1, pp. 281-306, 2017. doi:10.46586/tosc.v2017.i1.281-306.

[31]. M. Sinha and S. Dutta, “Survey on lightweight cryptography algorithm for data privacy in internet of things,” *Proc. Fourth International Conference on Microelectronics, Computing and Communication Systems*: MCCS 2019. Singapore: Springer, 2021.

[32]. M. Tolba et al., “Multidimensional zero-correlation linear cryptanalysis of reduced round SPARX-128,” *Selected Areas in Cryptography–SAC 2017*, Revised Selected Papers

24: 24th International Conference, Ottawa, ON, Canada, August 16-18, 2017. Springer International Publishing, 2018.

[33]. K. Sakan, et al., "Development and analysis of the new hashing algorithm based on block cipher," *East. Eur. J. Enterpr. Technol.*, vol. 2, no. 9 (116), p. 60-73, 2022. doi:10.15587/1729-4061.2022.252060.

[34]. W. J. Buchanan and L. Maglaras, *'Review of the NIST Light-weight Cryptography Finalist.' arXiv Preprint ArXiv:2303.14785*, 2023.

[35]. J.-P. Kaps, "Chai-tea, cryptographic hardware implementations of xtea," *Prog. Cryptol.-INDOCRYPT*, *Proc. 9*: 9th International Conference on Cryptology in India, Kharagpur, India, December 14-17, 2008. Berlin Heidelberg: Springer, *2008*.

[36]. Z. Mishra and B. Acharya, "Efficient hardware implementation of TEA, XTEA and XXTEA lightweight ciphers for low resource IoT applications," *Int. J. High Perform. Syst. Archit.*, vol. 10, no. 2, pp. 80-88, 2021. doi:10.1504/IJHPSA.2021.119150.

[37]. V. Prakash et al., "A new model of light weight hybrid cryptography for internet of things" 3rd International conference on Electronics, Communication and Aerospace Technology (ICECA). IEEE, *2019*. doi:10.1109/ICECA.2019.8821924.

[38]. A. Poojary et al., "FPGA implementation novel lightweight MBRISI cipher," *J. Ambient Intell. Hum. Comput.*, pp. 1-13, 2022.

[39]. M. Imdad et al., "An enhanced key schedule algorithm of PRESENT-128 block cipher for random and non-random secret keys," *Symmetry*, vol. 14, no. 3, p. 604, 2022. doi:10.3390/sym14030604.