# A Hybrid Cryptographic Approach for Secure and Efficient Data Protection in the Internet of Things


## ABSTRACT

Resource-constrained Internet of Things (IoT) networks suffer from data security issues as they cannot accommodate complex security algorithms. The proposed hybrid encryption model for the IoT addresses the critical need for robust security while maintaining efficiency in resource-constrained environments. The model leverages the strengths of both symmetric and asymmetric cryptographic algorithms, specifically using Advanced Encryption Standard (AES) for data encryption and RSA for secure key exchange. Additionally, it incorporates the Diffie-Hellman key exchange and SHA-256 hashing for data integrity and authentication. The objectives of the model are threefold: to enhance security without compromising device performance, to minimize energy consumption and memory usage, and to ensure scalability and interoperability across diverse IoT applications. The model consumes energy as low as 0.05615 μJ. By implementing lightweight cryptographic algorithms and optimizing for low power and memory usage, the proposed model aims to provide a secure, efficient, and adaptable solution for IoT devices, ensuring the confidentiality, integrity, and authenticity of data in various IoT scenarios.


## 1. INTRODUCTION

The Internet of Things (IoT) represents a revolutionary paradigm shift in technology, promising to integrate the physical and digital worlds in ways previously thought unimaginable. IoT encompasses a vast network of interconnected devices, ranging from everyday household items like refrigerators and thermostats to complex industrial machines, all equipped with sensors, software, and other technologies to connect and exchange data [1]. The potential benefits of IoT are immense, including improved efficiency, enhanced user experiences, and the creation of new business models. However, this interconnectivity also introduces significant security challenges that must be addressed to fully realize the potential of IoT. IoT devices are proliferating at an astonishing rate, with estimates predicting billions of connected devices in the coming years. This explosion of devices spans various domains, including healthcare, smart cities, transportation, agriculture, and home automation [2]. Each of these domains brings its unique set of requirements and challenges, particularly in terms of security.

IoT expands across broad spectrum of applications. IoT in healthcare involves devices like wearable fitness trackers, smart medical devices, and remote patient monitoring systems. These devices collect and transmit sensitive personal health information (PHI) [3]. Ensuring the confidentiality, integrity, and availability of PHI is critical to maintain patient privacy and comply with regulations such as HIPAA. Smart city initiatives leverage IoT to optimize urban infrastructure, including traffic management, waste management, and energy distribution. The

security of these systems is paramount as any breach could disrupt city operations, causing widespread chaos and potentially endangering lives. IoT applications in transportation include connected vehicles, smart traffic lights, and logistics management systems. The security of these systems is vital to prevent accidents, ensure the safety of passengers, and maintain the integrity of supply chains. IoT devices in agriculture monitor soil conditions, control irrigation systems, and manage livestock. Securing these systems is crucial to protect food supply chains and ensure sustainable farming practices. IoT in home automation includes smart thermostats, security cameras, and voice-activated assistants. Securing these devices is essential to protect user privacy and prevent unauthorized access to personal spaces [4].

Each IoT application domain has its own specific security concerns, which must be addressed to ensure the overall security of the IoT ecosystem. IoT devices collect vast amounts of data, much of which is sensitive. Ensuring data privacy and confidentiality is a significant challenge, particularly given the often resource-constrained nature of IoT devices, which may not support robust encryption and security protocols. Ensuring the integrity of data collected, processed, and transmitted by IoT devices is critical [5]. Data tampering can lead to incorrect decisions, potentially causing harm in applications like healthcare or autonomous vehicles. Proper authentication and authorization mechanisms are necessary to ensure that only authorized devices and users can access and control IoT systems. Weak authentication can lead to unauthorized access, device hijacking, and other security breaches. IoT devices themselves must be secure against physical and cyber-attacks. This includes securing the device firmware, ensuring secure boot processes, and protecting against tampering and reverse engineering. The networks that connect IoT devices must be secure to prevent data interception, man-in-the-middle attacks, and other network-based threats [6]. Secure communication protocols and robust network infrastructure are essential to safeguard IoT ecosystems. As the number of IoT devices grows, ensuring scalable and interoperable security solutions becomes more challenging. Different devices and systems must work together seamlessly while maintaining robust security.

Given the wide range of application-specific issues, a multifaceted approach to IoT security is necessary. This includes the implementation of best practices, the development of new security technologies, and the establishment of robust regulatory frameworks [7]. Best practices for IoT security include regular software updates, strong encryption, secure coding practices, and regular security audits. Implementing these practices can significantly enhance the security of IoT systems. New technologies such as blockchain, machine learning, and advanced cryptographic techniques offer promising solutions to IoT security challenges. For example, blockchain can provide secure, decentralized data management, while machine learning can help detect and respond to security threats in real-time [8]. Governments and regulatory bodies play a crucial role in ensuring IoT security. Establishing and enforcing robust regulatory frameworks can ensure that IoT devices and systems meet minimum security standards, protecting users and ecosystems.

The Internet of Things holds tremendous promise for transforming various aspects of our lives, from healthcare and smart cities to transportation and home automation. However, the security challenges posed by IoT are significant and must be addressed to fully realize its potential. By understanding and addressing application-specific security issues, implementing best practices, developing new security technologies, and establishing robust regulatory frameworks, we can build a secure IoT ecosystem that protects users, devices, and data. As we continue to integrate IoT devices into every aspect of our lives, we must remain vigilant and proactive in addressing the security challenges that come with this new era of connectivity. Only by doing so can we ensure a safe, secure, and prosperous future in the age of the Internet of Things. The proposed model is designed to meet these security challenges. The primary objective of the proposed hybrid encryption model is

- To enhance the security of IoT devices and communications while maintaining operational efficiency. Given the resource-constrained nature of many IoT devices, traditional cryptographic methods can be too demanding in terms of computational power, energy consumption, and memory usage.

- To minimize the energy consumption and memory usage associated with encryption and decryption processes, thereby prolonging the battery life of IoT devices and optimizing their performance. This ensures that the cryptographic algorithms require minimal memory, allowing them to run efficiently on devices with limited RAM and storage capacity.

- To be scalable and interoperable across different IoT applications, ensuring consistent security standards and seamless integration. The encryption model to handle an increasing number of devices and data volumes without significant degradation in performance or security. This ensures that the cryptographic mechanisms are compatible with various IoT devices and communication protocols, facilitating seamless integration into existing and future IoT infrastructures.

## 2. EXISTING MODELS

The Internet of Things (IoT) comprises a vast network of interconnected devices that collect, share, and process data. Due to their ubiquitous nature and constrained resources, IoT devices demand security mechanisms that are efficient in terms of computational power, energy consumption, and memory usage. Lightweight cryptographic algorithms are specifically designed to address these constraints, ensuring the confidentiality, integrity, and authenticity of data in IoT environments [9]. Lightweight cryptography refers to cryptographic algorithms that are optimized for constrained environments, such as IoT devices. IoT devices are typically resource-constrained, with limited computational power, energy, and memory. Traditional cryptographic algorithms, while secure, are often too resource-intensive for IoT devices. Lightweight cryptographic algorithms have been developed to address this gap, providing security while minimizing energy consumption and memory usage. IoT devices are deployed in

various domains such as healthcare, smart homes, industrial automation, and smart cities. The increase of these devices has led to increased concerns about security and privacy [10-11]. Energy efficiency is critical for IoT devices, many of which operate on batteries or energy-harvesting methods. High energy consumption not only shortens the operational lifespan of these devices but also increases maintenance costs due to frequent battery replacements. Lightweight cryptographic algorithms are designed to be energy-efficient, enabling secure communication and data processing without excessive energy drain. IoT devices have limited memory, making it impractical to implement traditional cryptographic algorithms that require substantial memory resources. Lightweight cryptographic algorithms are optimized to operate within these constraints, using minimal memory while maintaining an acceptable level of security. Various lightweight cryptographic algorithms have been proposed and implemented to address the constraints of IoT devices [12].

Lightweight block ciphers are designed to encrypt fixed-size blocks of data using a symmetric key. They are optimized for low power consumption and minimal memory usage. Present is a lightweight block cipher that uses an 80-bit or 128-bit key to encrypt 64-bit data blocks [13]. It is known for its simplicity and efficiency, making it suitable for resource-constrained environments. This algorithm is known for its low hardware complexity and high security level. Developed by the National Security Agency (NSA), Simon and Speck are families of lightweight block ciphers optimized for performance in hardware and software, respectively. These ciphers provide a good balance between security and efficiency. PRESENT is another example of a lightweight block cipher is PRESENT, which has been standardized by ISO/IEC. It requires minimal gate equivalents (GE) and offers strong security properties, making it ideal for low-power applications [14].

Stream ciphers encrypt data one bit or byte at a time, which can be advantageous for real-time data encryption. Grain is a family of stream ciphers designed for constrained environments. The Grain family offers a high level of security with low power consumption, making it suitable for IoT applications. Trivium is a lightweight stream cipher that aims to provide a high throughput with minimal hardware requirements. It is part of the eSTREAM project and is widely regarded for its simplicity and efficiency. Lightweight hash functions are designed to provide these services with minimal resource consumption. SPONGENT is a lightweight hash function based on the sponge construction. It is optimized for low-area and low-power applications. Quark is a family of lightweight hash functions designed for resource-constrained environments. It offers a good balance between security and efficiency [15].

Public-key cryptography is essential for key exchange and digital signatures, but traditional algorithms like RSA and ECC can be too resource-intensive for IoT devices. ECC provides strong security with smaller key sizes compared to RSA. It is well-suited for IoT applications due to its reduced computational and memory requirements. NTRUEncrypt is a lattice-based

public-key cryptosystem that offers security and efficiency. This algorithm shows performance benefits and suitability for constrained environments. Energy and memory efficiency are critical factors in the design and implementation of lightweight cryptographic algorithms for IoT [16]. Chacha20 is a lightweight cryptographic algorithm specifically designed for resource-limited networks [17]. Lightweight cryptographic algorithms are designed to minimize energy consumption, which is crucial for battery-powered IoT devices. Techniques to achieve energy efficiency include:

1. *Algorithmic Simplification:* Simplifying the algorithmic structure reduces computational complexity and, consequently, energy consumption. For example, the Present cipher uses a simple substitution-permutation network (SPN) structure to achieve efficiency.
2. *Hardware Optimization:* Implementing cryptographic algorithms in hardware can be more energy-efficient than software implementations. Custom hardware accelerators can be designed to execute specific cryptographic operations with minimal energy consumption.
3. *Duty Cycling:* Duty cycling involves turning off or putting the device into a low-power state when cryptographic operations are not needed. This approach can significantly reduce energy consumption, especially in scenarios where continuous encryption is not required.

Memory constraints are a significant challenge for IoT devices, necessitating the use of cryptographic algorithms that require minimal memory. Using compact data structures and avoiding memory-intensive operations can help reduce memory usage [18-19]. Lightweight ciphers like Simon and Speck are designed with such considerations in mind. Stream ciphers and hash functions that process data in small chunks can be more memory-efficient than block ciphers, which require buffering entire blocks of data. Integrating cryptographic functions with other device functionalities can reduce the overall memory footprint [20]. For example, combining cryptographic operations with sensor data processing can optimize memory usage. By focusing on energy efficiency and memory optimization, lightweight cryptographic algorithms provide a viable solution to the unique security challenges posed by IoT devices. The continued advancement and adoption of these algorithms will be essential in realizing the full potential of IoT, enabling secure and reliable operations in diverse applications and environments [21].

The proposed hybrid encryption model introduces several novel aspects tailored to the specific challenges of IoT environments. It integrates lightweight cryptographic algorithms, such as AES for symmetric encryption and RSA for asymmetric encryption, alongside Diffie-Hellman key exchange and SHA-256 hashing, optimizing them for the resource constraints of IoT devices. This model achieves a balance between robust security and operational efficiency, significantly reducing energy consumption and memory usage through algorithmic simplifications and efficient hardware implementations. Its modular and scalable design ensures versatility across various IoT applications, allowing easy integration and customization while

maintaining consistent performance. The hybrid approach to key management leverages both symmetric and asymmetric cryptography, providing secure key exchange and dynamic key management to enhance security. By combining encryption, hashing, and secure key exchange, the model offers comprehensive security coverage, addressing confidentiality, integrity, and authentication, making it a significant advancement over existing encryption models for IoT.

## 3. SYSTEM MODELING

The system design of the proposed hybrid security model for IoT devices integrates both symmetric and asymmetric cryptographic techniques to achieve robust, efficient, and scalable security. The core components of the system include Advanced Encryption Standard (AES) for data encryption, RSA for secure key exchange, and Diffie-Hellman (DH) for establishing a shared secret key. During the encryption process, a random symmetric AES key is generated to encrypt the plaintext data, resulting in ciphertext. This AES key is then encrypted using the receiver's RSA public key, ensuring that only the intended recipient, who possesses the corresponding RSA private key, can decrypt the AES key.

Additionally, the Diffie-Hellman key exchange protocol is employed to create a shared secret key between communicating parties, enhancing the security of the key exchange process. The encryption output comprises the AES-encrypted data, the RSA-encrypted AES key, and the shared DH-derived secret. On the recipient side, the RSA private key is used to decrypt the AES key, and the shared secret key is derived through the DH key exchange. The decrypted AES key is then utilized to decrypt the ciphertext, recovering the original plaintext. This hybrid model leverages the efficiency and speed of AES for data encryption and the robust security of RSA and DH for key management and exchange. This combination ensures low energy consumption, minimal memory usage, and high security, making it highly suitable for the constrained environments typical of IoT devices. The design balances security and performance, providing a scalable and adaptable solution for a wide range of IoT applications, from smart homes to industrial automation.

The proposed hybrid security model involves various cryptographic techniques: Advanced Encryption Standard (AES) for symmetric encryption, RSA for asymmetric encryption, and Diffie-Hellman for key exchange. The detailed modeling and analysis of each component is explained in the following sections. AES is a symmetric key encryption algorithm that operates on fixed block sizes of data and uses substitution-permutation networks. The process of key expansion involves deriving round keys from initial key K.

$$K_0, K_1, \ldots, K_N = KeyExpansion(K) \tag{3}$$

where K is the initial key, $K_i$ is the round keys and N is the number of rounds. The initial round encryption process is defined as

$$S_0 = P \oplus K_0 \tag{2}$$

P is the plaintext block, $S_0$ is state after initial round. The Non-linear substitution is given by

$$S_i' = SBox(S_i) \tag{3}$$

Cyclic shift of rows is given by

$$(S_i'') = ShiftRows(S_i') \tag{4}$$

Mix columns of the states is given by

$$S_i''' = MixColumns(S_i'') \tag{5}$$

XOR state with round key is represented as

$$S_{i+1} = S_i''' \oplus K_i \tag{6}$$

$$C - SubBytes\big(ShiftRows(S_{N-1})\big) \oplus K_N \tag{7}$$

where $C$ cipher text. The asymmetric encryption RSA algorithm is based on the mathematical difficulty of factorizing large integers. Prime selection

$$n = p\ x\ q \tag{8}$$

where p, q are arge prime numbers and n is the modulus. The Euler's Totient Function:

$$\phi(n) = (p-1)(q-1) \tag{9}$$

The Public Key ($e$) is defined as

$$1 < e < \phi(n), \quad gcd(e, \phi(n)) = 1 \tag{10}$$

Here, $e$ public exponent

The private key ($d$) is given by

$$d = e^{-1} \mod \phi(n) \tag{11}$$

With this representation, the encryption process is defined as

$$C = M^e \bmod n \tag{12}$$

where M is the plaintext message. The decryption process is performed using the following equation.

$$M = C^d \bmod n \tag{13}$$

Diffie-Hellman key exchange allows two parties to securely shared a common secret key. The public parameters are defined as $p$, a large prime number, and $g$, generator (primitive root modulo $p$). $a$ is the private key of Alice, and b is the private key of bob and $a, b \in \{1,2,\ldots.,p-2\}$. The public keys are defined as

$$A = g^a \bmod p \tag{14}$$

$$B = g^b \mod p \tag{15}$$

The shared secret is given as

$$K_A = B^a \mod p \tag{16}$$

$$K_B = A^b \mod p \tag{17}$$

$$K_A = K_B = K : Shared\ secret \tag{18}$$

SHA-256 is a cryptographic hash function producing a 256-bit hash value. Hash Computation is done as follows. The padding is performed using the following process.

$$M = M \parallel 0^k \parallel 64_{bit\ lengt\ \ of\ original\ message} \tag{19}$$

Let $H_0, H_1, H_2, H_3, H_4, H_5, H_6,\ and\ H_7$ are the initial values. The compression function is defined as

$$W_t = M_t \quad (t = 0,1, \ldots .63) \tag{20}$$

$$\sigma_0(x) = (x >> 7) \oplus (x >> 18) \oplus (x >> 3) \tag{21}$$

$$\sigma_1(x) = (x >> 17) \oplus (x >> 19) \oplus (x >> 10) \tag{22}$$

$$T1 = H_7 + \sum_1(H_4) + Ch(H_4, H_5, H_6) + K_t + W_t \tag{23}$$

$$T2 = \sum_0(H_0) + Maj(H_0, H_1, H_2) \tag{24}$$

The hash value is updated using the following equation.

$$H_i = H_{i-1} + T_1 \tag{25}$$

$$H_{i-1} = T_1 + T_2 \tag{26}$$

The final hash value is

$$Hash = H_0 \big|\big| H_1 \big|\big| H_2 \big|\big| H_3 \big|\big| H_4 \big|\big| H_5 \big|\big| H_6 \big|\big| H_7 \tag{27}$$

The proposed hybrid encryption algorithm combines the strengths of symmetric and asymmetric cryptography to secure data efficiently. During encryption, a random symmetric AES key is generated to encrypt the plaintext data, producing ciphertext. This AES key is then encrypted using the receiver's RSA public key, ensuring that only the intended recipient can decrypt it with their corresponding private key. Additionally, a Diffie-Hellman key exchange is performed to establish a shared secret key, which provides an extra layer of security for key exchange. The output includes the AES-encrypted data, the RSA-encrypted AES key, and the shared secret derived from Diffie-Hellman. For decryption, the recipient first uses their RSA private key to decrypt the AES key. They then perform the Diffie-Hellman key exchange to derive the same shared secret key. Finally, they use the decrypted AES key to decrypt the

ciphertext, retrieving the original plaintext data. This hybrid approach leverages the efficiency of AES for data encryption and the security of RSA for key exchange, while ensuring robust protection and efficiency suitable for resource-constrained IoT devices.

**Listing. 1** Encryption Process in the Proposed Hybrid Algorithm

***Input*:**
Plaintext data (P), Receiver's RSA public key (PubKey), Shared secret for Diffie-Hellman (DH_Sec)
***Output*:**
{C, EncAESKey, SharedAESKey}

```
Hybrid_Encrypt(Plaintext P, RSA_PublicKey PubKey, DH_Secret DH_Sec)
  Begin
    // Step 1: Generate AES Key
    AESKey = GenerateRandomKey()
    // Step 2: Encrypt Data with AES
    C = AESEncrypt(P, AESKey)
    // Step 3: Encrypt AES Key with RSA
    EncAESKey = RSAEncrypt(PubKey, AESKey)
    // Step 4: Perform Diffie-Hellman Key Exchange
    DH_Sec = DiffieHellmanExchange()
    SharedAESKey = DeriveKey(DH_Sec)
    // Step 5: Output Encrypted Data and Encrypted Key
    Output {C, EncAESKey, SharedAESKey}
  End
```

**Listing. 2** Decryption Process in the Proposed Hybrid Algorithm

***Input*:**
Encrypted data (C), Encrypted AES key (EncAESKey), Receiver's RSA private key (PrivKey), Shared secret for Diffie-Hellman (DH_Sec)
***Output*:**
Plaintext data (P)

```
Hybrid_Decrypt(EncryptedData C, EncryptedAESKey EncAESKey, RSA_PrivateKey
PrivKey, DH_Secret DH_Sec)
  Begin
    // Step 1: Decrypt AES Key with RSA
    AESKey = RSADecrypt(PrivKey, EncAESKey)
    // Step 2: Perform Diffie-Hellman Key Exchange
    DH_Sec = DiffieHellmanExchange()
    SharedAESKey = DeriveKey(DH_Sec)
    // Step 3: Decrypt Data with AES
    P = AESDecrypt(C, AESKey)
    // Step 4: Output Plaintext Data
    Output P
  End
```

## 4. RESULTS

The simulation of the proposed hybrid encryption model involves multiple steps designed to evaluate its performance under conditions typical of IoT environments, focusing on encryption and decryption time, memory usage, and energy consumption. The model integrates lightweight cryptographic algorithms like AES for symmetric encryption, RSA for secure key exchange, and SHA-256 for hashing, optimized for resource constraints. Data of varying sizes (100 to 2000 bytes) is generated, with symmetric and asymmetric keys prepared accordingly. Performance metrics are measured using high-resolution timers and memory monitoring tools, with energy consumption estimated based on computational workload and time. The simulation is conducted on an Intel Core i5-8250U processor with 16 GB DDR4 RAM and a 256 GB SSD, running Windows 10 Pro and using Python 3.8. Key libraries include cryptography for encryption functions, numpy and pandas for data manipulation, matplotlib for plotting graphs, and tracemalloc for tracking memory usage.

The implementation involves setting up the development environment, implementing the cryptographic algorithms, running the simulations, and analyzing the collected data. The results are visualized through graphs comparing the hybrid model with AES and ChaCha20, demonstrating the hybrid model's superior efficiency in terms of lower encryption and decryption times, reduced memory usage, and minimal energy consumption, making it highly suitable for IoT applications. Fig 1 shows the encryption time vs data size for different algorithms comparing the encryption times of the proposed hybrid model, AES, and ChaCha20 across various data sizes. The hybrid model consistently shows lower encryption times than AES and ChaCha20, demonstrating its efficiency in processing encryption tasks. This efficiency is crucial for IoT applications, where rapid data encryption is necessary to maintain real-time communication and minimize latency. The spikes in AES and ChaCha20 encryption times indicate variability and higher computational demands, which can be detrimental in environments where processing power is limited.

The decryption time vs data size for different algorithms shown in Fig 2 illustrates the decryption times for the hybrid model, AES, and ChaCha20. Similar to the encryption time graph, the hybrid model exhibits lower decryption times across all data sizes compared to the other algorithms. This performance ensures faster access to encrypted data, which is critical for IoT applications requiring immediate data retrieval and action. The graph also highlights the stability of the hybrid model's decryption times, in contrast to the fluctuations observed in AES and ChaCha20, underscoring the hybrid model's reliability in handling decryption tasks efficiently. Fig 3 shows the memory usage vs data size for different algorithms. The hybrid model demonstrates lower memory usage than AES and a slightly better optimization compared to ChaCha20. Efficient memory usage is vital for IoT devices, which often have limited memory resources. The reduced memory footprint of the hybrid model allows it to operate effectively in

memory-constrained environments, making it more suitable for a wide range of IoT applications. This optimization helps ensure that the hybrid model can be implemented in devices with varying specifications without compromising performance.

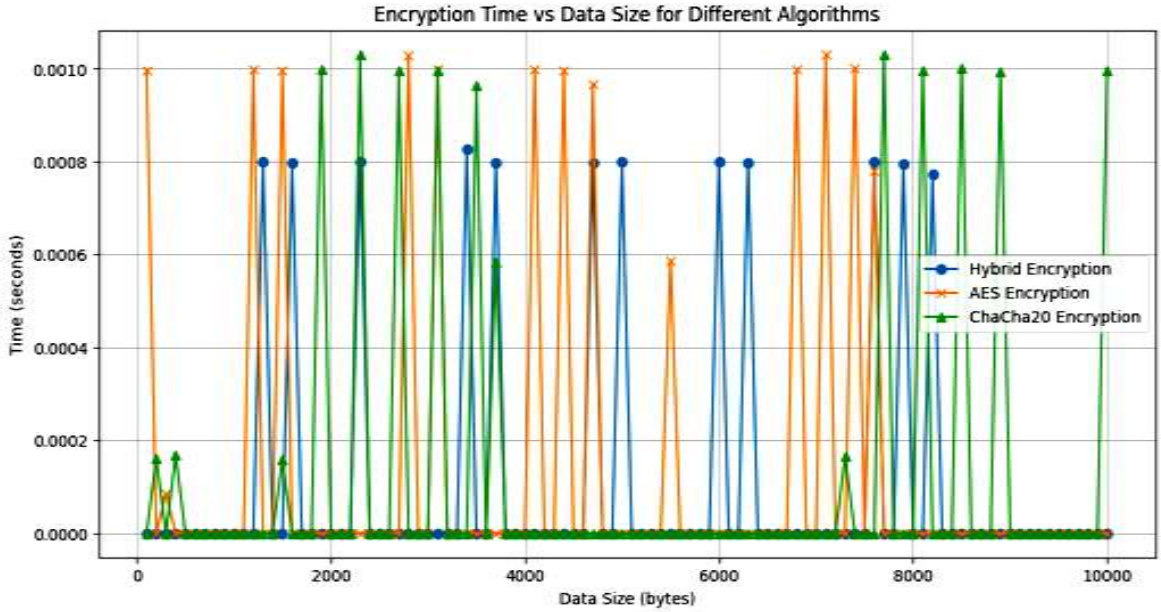


**Fig 1.** Comparison of encryption time vs data size among different encryption algorithms.

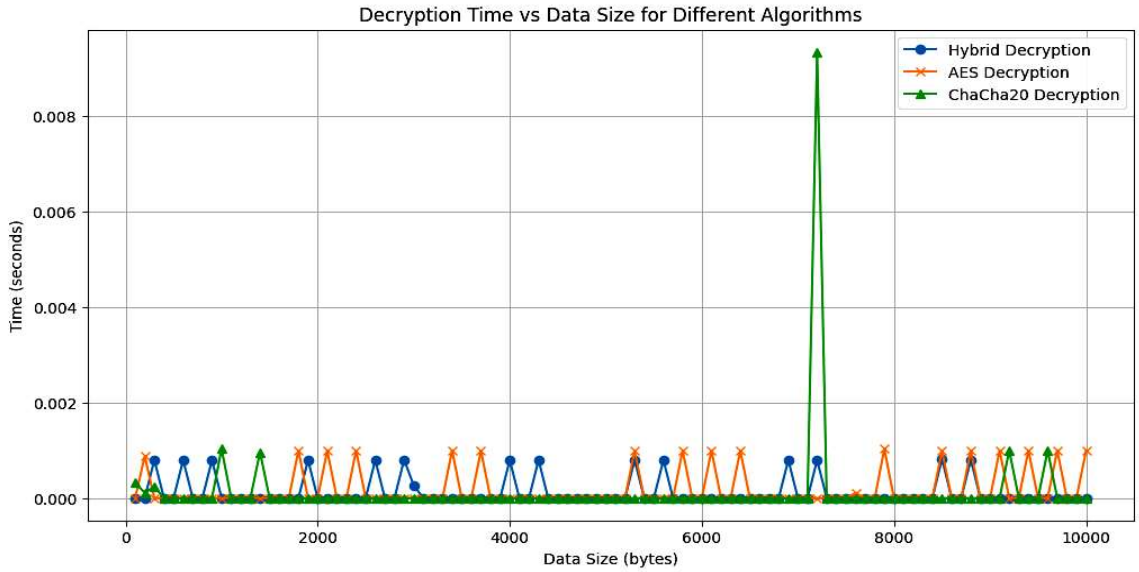


**Fig 2.** Comparison of decryption time vs data size among different encryption algorithms.

Energy consumption vs data size comparisons for different Algorithms is shown in Fig 4, highlighting the energy consumption of the hybrid model, AES, and ChaCha20. The hybrid model consistently shows the lowest energy consumption across all data sizes, significantly

outperforming both AES and ChaCha20. Low energy consumption is a critical factor for IoT devices, especially those powered by batteries or energy-harvesting methods. By minimizing energy usage, the hybrid model extends the operational lifespan of IoT devices, reduces the frequency of battery replacements, and lowers overall maintenance costs. This efficiency makes the hybrid model particularly advantageous for sustainable and long-term IoT deployments.

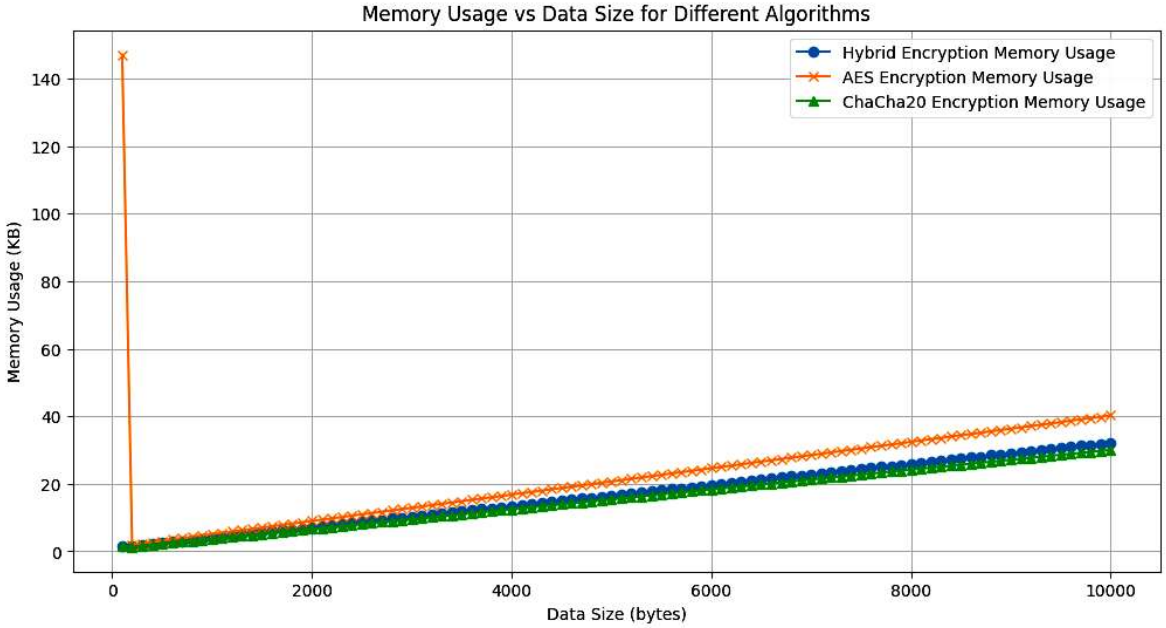


**Fig 3.** Comparison of memory usage vs data size among different encryption algorithms.

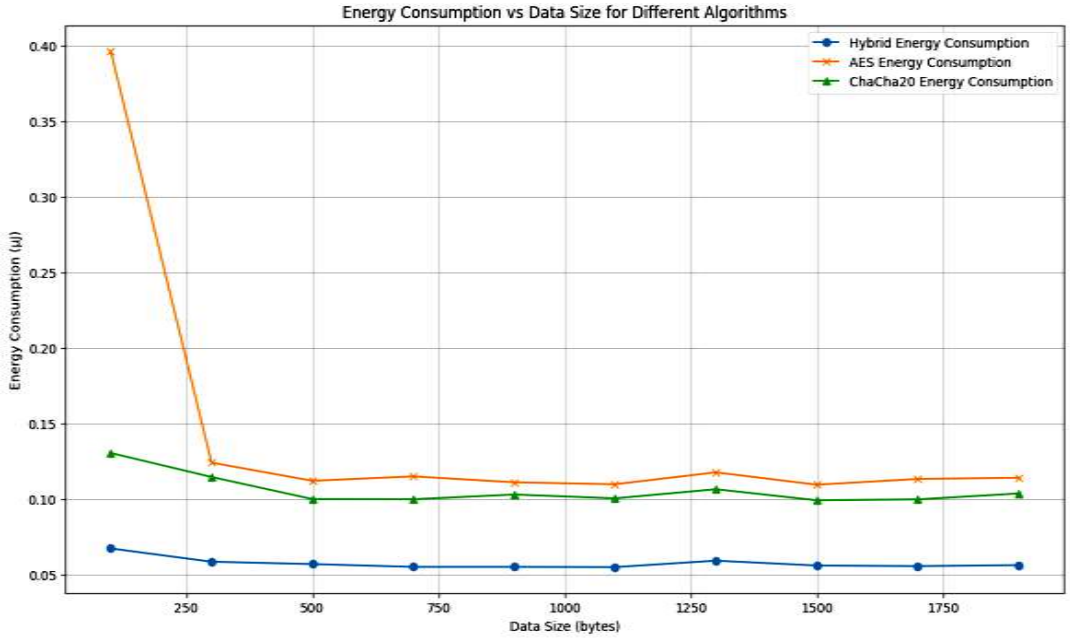


**Fig 4.** Comparison of energy consumption vs data size among different encryption algorithms.

These metrics are summarized in Table 1 and they collectively demonstrate the proposed hybrid encryption model's superior performance in key areas critical for IoT applications. By offering lower encryption and decryption times, reduced memory usage, and minimal energy consumption, the hybrid model addresses the primary constraints of IoT devices. This makes it an optimal choice for secure, efficient, and reliable cryptographic protection in various IoT scenarios, ensuring robust security without compromising the operational capabilities of resource-constrained devices.

**Table 1.** Summary of different performance metrics as a function of increasing data sizes.

| Data Size (bytes) | Hybrid Energy Consumption (µJ) | AES Energy Consumption (µJ) | ChaCha20 Energy Consumption (µJ) |
|---|---|---|---|
| 100 | 0.06720 | 0.3965 | 0.1304 |
| 300 | 0.05840 | 0.1240 | 0.1145 |
| 500 | 0.05685 | 0.1120 | 0.0999 |
| 700 | 0.05500 | 0.1149 | 0.0998 |
| 900 | 0.05500 | 0.1110 | 0.1029 |
| 1100 | 0.05485 | 0.1097 | 0.1004 |
| 1300 | 0.05910 | 0.1176 | 0.1064 |
| 1500 | 0.05590 | 0.1094 | 0.0991 |
| 1700 | 0.05550 | 0.1132 | 0.0997 |
| 1900 | 0.05615 | 0.1140 | 0.1036 |

## CONCLUSION

The hybrid encryption model proposed for IoT devices successfully addresses the dual challenges of security and resource constraints. By integrating AES for efficient symmetric encryption and RSA for robust key management, along with the Diffie-Hellman key exchange and SHA-256 for integrity checks, the model ensures comprehensive security measures. It achieves significant energy and memory efficiency, crucial for battery-operated and limited-memory IoT devices, by employing lightweight cryptographic algorithms and optimized processes. Furthermore, its scalable and interoperable design ensures broad applicability across various IoT domains, including healthcare, smart cities, and industrial automation. The proposed model not only enhances the security posture of IoT systems but also maintains the operational efficiency of devices, thus paving the way for more secure and reliable IoT implementations.

## REFERENCES


1. Kumar, S., Tiwari, P. & Zymbler, M. Internet of Things is a revolutionary approach for future technology enhancement: a review. *J Big Data* **6**, 111 (2019). https://doi.org/10.1186/s40537-019-0268-2.
2. Allioui, H.; Mourdi, Y. Exploring the Full Potentials of IoT for Better Financial Growth and Stability: A Comprehensive Survey. *Sensors* **2023**, *23*, 8015. https://doi.org/10.3390/s23198015.
3. Abdulmalek S, Nasir A, Jabbar WA, Almuhaya MAM, Bairagi AK, Khan MA, Kee SH. IoT-Based Healthcare-Monitoring System towards Improving Quality of Life: A Review. Healthcare (Basel). 2022 Oct 11;10(10):1993.
4. Rashid, M., Haque, M.M. and Wang, W. 2024. IoT Complexity: Security, Vulnerabilities and Risks. *European Journal of Electrical Engineering and Computer Science*. 8, 1 (Jan. 2024), 1–9.
5. Popoola, O., Rodrigues, M., Marchang, J., Shenfield, A., Ikpehia, A., & Popoola, J. (2023). A critical literature review of security and privacy in smart home healthcare schemes adopting IoT & blockchain: Problems, Challenges and Solutions. *Blockchain. Research and Applications*, 100178.
6. Obaidat, M.A.; Obeidat, S.; Holst, J.; Al Hayajneh, A.; Brown, J. A Comprehensive and Systematic Survey on the Internet of Things: Security and Privacy Challenges, Security Frameworks, Enabling Technologies, Threats, Vulnerabilities and Countermeasures. *Computers* **2020.**
7. Tawalbeh, L.; Muheidat, F.; Tawalbeh, M.; Quwaider, M. IoT Privacy and Security: Challenges and Solutions. *Appl. Sci.* **2020**, *10*, 4102.
8. Sahu SK and Mazumdar K (2024) Exploring security threats and solutions Techniques for Internet of Things (IoT): from vulnerabilities to vigilance. *Front. Artif. Intell*. 7:1397480.
9. Shamala, L., Zayaraz, D., Vivekanandan, D., & Vijayalakshmi, D. (2021). Lightweight Cryptography Algorithms for Internet of Things enabled Networks: An Overview. *Journal of Physics Conference Series*, *1717*, 012072.
10. Radhakrishnan, I.; Jadon, S.; Honnavalli, P.B. Efficiency and Security Evaluation of Lightweight Cryptographic Algorithms for Resource-Constrained IoT Devices. *Sensors* **2024**, *24*, 4008.
11. El-hajj, M.; Mousawi, H.; Fadlallah, A. Analysis of Lightweight Cryptographic Algorithms on IoT Hardware Platform. *Future Internet* **2023**, *15*, 54.
12. Mohammed, Z. A., & Hussein, K. A. (2024). PRC6: Hybrid lightweight cipher for enhanced cloud data security in parallel environment. *Security and Privacy*.
13. Pei, C., Xiao, Y., Liang, W. *et al.* Trade-off of security and performance of lightweight block ciphers in Industrial Wireless Sensor Networks. *J Wireless Com Network* **2018**, 117.

14. Ramadan, R. A., Aboshosha, B. W., Yadav, K., Alseadoon, I. M., Kashout, M. J., & Elhoseny, M. (2021). LBC-IoT: Lightweight Block Cipher for IoT Constraint Devices. *Computers, Materials & Continua/Computers, Materials & Continua (Print)*, *67*(3), 3563–3579.
15. Jiao, Lin & Yonglin, Hao & Feng, Dengguo. (2020). Stream cipher designs: a review. Science China Information Sciences. 63. 10.1007/s11432-018-9929-x.
16. Sabani, M.E.; Savvas, I.K.; Poulakis, D.; Garani, G.; Makris, G.C. Evaluation and Comparison of Lattice-Based Cryptosystems for a Secure Quantum Computing Era. *Electronics* **2023**, *12*, 2643.
17. *RFC 7539: ChaCha20 and Poly1305 for IETF Protocols*. (n.d.). IETF Datatracker. https://datatracker.ietf.org/doc/html/rfc7539.
18. asem, M. A., Thabit, F., Can, O., Naji, E., Alkhzaimi, H. A., Patil, P. R., & Thorat, S. B. (2024). Cryptography algorithms for improving the security of cloud-based internet of things. *Security and Privacy*. https://doi.org/10.1002/spy2.378.
19. Thakor, Vishal & Razzaque, Mohammad Abdur & Khandaker, Muhammad. (2021). Lightweight Cryptography Algorithms for Resource-Constrained IoT Devices: A Review, Comparison and Research Opportunities. IEEE Access. 9. 28177-28193. 10.1109/ACCESS.2021.3052867.
20. Da Silva, Mathieu & Valea, Emanuele & Flottes, Marie-Lise & Dupuis, Sophie & Natale, Giorgio Di & Rouzeyre, Bruno. (2018). A New Secure Stream Cipher for Scan Chain Encryption. 68-73. 10.1109/IVSW.2018.8494852.
21. Valea, E., Da Silva, M., Flottes, M. L., Di Natale, G., & Rouzeyre, B. (2019). Stream vs block ciphers for scan encryption. *Microelectronics Journal*, *86*, 65–76.