Hybrid Lightweight Cryptography for Data Security

Sriraam C  
School of Computer Science and Engineering  
Vellore Institute of TechnologyChennai, TamilNadu  
sriraam.c2021@vitstudent.ac.in

Balaji Ramesh   
School of Computer Science and Engineering  
Vellore Institute of TechnologyChennai, TamilNadu  
balaji.r2021@vitstudent.ac.in

Lionel Donato L   
School of Computer Science and Engineering  
Vellore Institute of TechnologyChennai, TamilNadu  
lioneldonato.l@vitstudent.ac.in

*Abstract*— Security is becoming important for both individuals and business needs. Security has been studied extensively by many researchers. Still, security gaps or threats are increasing, and data protection is a primary issue. A novel, effective lightweight hybrid cryptographic algorithm with two layers of encryption is presented in this paper. A new effective, lightweight cryptographic algorithm is used in the first layer, and a Dynamic RSA Technique is used in the second layer. Both symmetric and asymmetric cryptography features are offered by this approach. The proposed approach’s performance is evaluated using various metrics, including computational time, key sensitivity, statistical analysis, image histogram, and entropy change analysis. A high level of security and an apparent improvement in encryption execution time, memory usage, and throughput compared to widely used cryptographic systems were shown by the experimental findings.

Keywords - Security, RSA, Lightweight hybrid cryptography, ASCON-128

# Introduction

Data transfer applications are vital for communication and collaboration, making the security and integrity of transmitted information crucial. Traditional encryption techniques often struggle to balance security and efficiency, particularly in real-time data transfer. To address this, a novel hybrid cryptographic algorithm that combines lightweight and traditional methods is proposed. Both symmetric and asymmetric algorithms are uniquely integrated by our method, enhancing security and optimizing performance. The strategic combination leverages the strengths of both types to mitigate their limitations, delivering improved security and efficiency in data transfer scenarios.

Unlike conventional approaches that rely solely on either symmetric or asymmetric encryption, a layered approach is employed by our hybrid method. This approach utilizes symmetric encryption for its speed and efficiency in handling large data volumes, while asymmetric encryption provides robust key management and authentication. This dual-layer integration ensures a more secure and efficient encryption process, suitable for a variety of data transfer applications.

The first layer of our approach includes ASCON 128-bit encryption, a lightweight symmetric authenticated encryption algorithm known for its efficiency and robust security. ASCON’s design is made ideal for constrained environments, such as IoT devices, where resource efficiency is paramount. Additionally, the encryption process is adjusted by the dynamic RSA method as the second layer based on input data characteristics, improving encoding and decoding times. By storing pre-computed values of frequently occurring data, computational overhead is significantly reduced by the dynamic RSA.

Extensive performance evaluations demonstrate that computational time and time complexity are significantly improved by our algorithm compared to existing methods. Experimental results, derived from real-world data transfer scenarios, showcase a substantial improvement in time complexity and computational time, validating the practicality and robustness of our approach.

The introduction of this hybrid algorithm aims to advance the security and efficiency of data transfers. A versatile solution is offered for various applications, from file sharing to real-time communication, contributing to the progress of encryption technologies and fostering a safer digital environment. Furthermore, seamless integration with cloud services is enabled by our proposed method, creating a comprehensive ecosystem for secure and efficient data transfer. By leveraging the dynamic approach and the strengths of both encryption types, an effective solution to the security and performance challenges prevalent in data transfer applications is provided by our hybrid cryptographic algorithm.

By combining the benefits of ASCON's lightweight, efficient design with the robust security features of dynamic RSA, a significant step forward in cryptographic research and its practical applications is represented by our hybrid algorithm. This innovative fusion has the potential to address the critical needs in data transfer security, ensuring that both performance and protection are maintained without compromise.

This paper has been divided into 8 sections. Section 1 is meant to serve as an overview containing an introduction to the proposed algorithm. Different studies that are linked to the research are outlined in Section 2. The proposed algorithm is explained in an elaborate manner in Section 3. The simulation and implementation environments are described in Section 4. The findings and comments that can be drawn from the findings are summarized in Section 5. A comparative analysis is contained in Section 6. The CIA’s security analysis and accomplishments are described in Section 7. Other conclusions and recommendations for future research are suggested in the final section i.e. Section 8.

# Related Works

A novel lightweight homomorphic cryptographic algorithm specifically designed for cloud computing environments was introduced by Thabit, Fursan, et al., showcasing an innovative approach that marries efficiency with security to address the critical need for robust data protection in cloud services [1]. A significant advancement in the field of cryptography is represented by this work, offering a solution that is both efficient and secure for cloud-based applications.

A dynamic method aimed at reducing time complexity in RSA encryption and decryption processes was unveiled, highlighting the importance of adaptive encryption techniques in enhancing the practicality of RSA for real-time applications [2]. By introducing a dynamic approach, contributions are made by this research to the ongoing efforts to optimize cryptographic algorithms for better performance and security.

An exhaustive review conducted offered deep insights into lightweight block ciphers, detailing their design principles, security features, and performance characteristics [3]. This comprehensive review serves as a valuable resource for those seeking to understand the landscape of lightweight cryptographic solutions suitable for constrained environments, providing a foundation for further research and development in this area.

A novel data security algorithm that leverages genetics techniques and logical-mathematical functions was proposed, showcasing the potential of interdisciplinary methods in enhancing cryptographic security within cloud computing [4]. This innovative approach demonstrates how combining different fields of study can lead to breakthroughs in cryptography, offering new ways to protect sensitive data in cloud environments.

A comprehensive overview of research trends in RSA-based asymmetric cryptography techniques was provided, underscoring the pivotal role of RSA in securing digital communications and highlighting both developments and challenges within RSA cryptography [5]. The importance of staying informed about the latest trends and advancements in cryptographic techniques to ensure effective data protection strategies is emphasized by this overview.

A systematic review explored the contributions of homomorphic encryption within the healthcare industry, emphasizing its potential to enhance privacy and efficiency in secure data processing and analytics [6]. By focusing on the healthcare sector, the practical applications of homomorphic encryption are highlighted by this review, showcasing its benefits in maintaining patient confidentiality while enabling data analysis for improved healthcare outcomes.

A hybrid cryptography technique designed to bolster network security by combining elements of different cryptographic methods was proposed by researchers, demonstrating the benefits of integrating diverse security mechanisms [7]. A more robust defense against cyber threats is offered by this hybrid approach, illustrating the effectiveness of combining cryptographic primitives for enhanced security.

A survey focused on elliptic curve lightweight cryptography was conducted by authors C. A. Lara-Nino, A. Diaz-Perez, and M. Morales-Sandoval, highlighting its efficiency and security advantages for resource-constrained devices, and showcasing the potential of elliptic curves in cryptographic systems [8]. The importance of lightweight cryptography in providing secure communication channels for devices with limited computational power, such as those used in IoT applications, is underscored by this survey.

Innovative cryptographic solutions blending traditional and modern techniques were proposed by AbdElminaam, Diaa Salama to safeguard cloud computing environments against evolving threats, addressing the critical challenge of enhancing cloud security [9]. By integrating advanced cryptographic algorithms with established security protocols, this research aims to fortify cloud infrastructures against potential vulnerabilities, ensuring the integrity and confidentiality of stored data.

An efficient hybrid cryptography algorithm was introduced by authors Hoobi and Mayes M, designed to improve security across various digital platforms by combining cryptographic primitives for enhanced security [10]. The trend towards hybrid cryptographic solutions is exemplified by this algorithm, offering a balanced approach that leverages the strengths of different encryption methods to protect sensitive information across a wide range of applications.

The application of the RSA algorithm for securing data in cloud computing environments was explored in this study [11], emphasizing the importance of robust encryption techniques in protecting sensitive information stored in the cloud. The role of RSA encryption in cloud security is underscored by this study, highlighting its effectiveness in safeguarding data against unauthorized access and ensuring compliance with privacy regulations.

A performance evaluation comparing RSA, ElGamal, and Paillier partial homomorphic encryption algorithms was conducted by authors Mohammed, Saja J., and Dujan B. Taha, offering valuable insights into efficiency and security trade-offs among these algorithms [12]. Through this comparative analysis, guidance is aimed to be provided to practitioners in selecting the most suitable cryptographic solutions for specific applications, considering factors such as computational efficiency and security requirements.

The necessity of integrating security measures with timing considerations in safety-critical industrial cyber-physical systems was discussed in the study [13], emphasizing the importance of timing predictability alongside security. The multifaceted nature of security in industrial applications is highlighted by this discussion, where both the integrity of data and the timely execution of operations are considered paramount.

The combination of compression and cryptography techniques for enhancing data security was investigated by authors Sharma, Ruchita, and Swarnalata Bollavarapu [14], exploring how compression can optimize storage and transmission security alongside encryption. The potential synergies between data compression and encryption are demonstrated by this innovative approach, offering a dual-layer protection mechanism that conserves storage space while maintaining data confidentiality.

Improvements to the RSA cryptographic system were proposed by the authors of [15], aiming to enhance its security and efficiency, contributing to ongoing efforts to refine RSA, a foundational element of modern cryptography. The evolving needs of digital communication are addressed by these improvements, ensuring that RSA remains a viable option for securing sensitive information in an increasingly complex technological landscape.

The authors Dobraunig, Christoph, et al. study introduced Ascon v1.2, a lightweight authenticated encryption and hashing algorithm [16], marking a significant contribution to the field of cryptography. This algorithm is particularly notable for its efficiency and security, making it well-suited for applications in memory constrained environments.

Xoodyak, a lightweight cryptographic scheme designed for efficiency and security in constrained environments, was introduced by Daemen, Joan, et al, addressing the growing need for cryptographic solutions that are both powerful and resource-friendly [17]. This introduction of Xoodyak represents a significant advancement in lightweight cryptography, catering to the demands of modern applications that require secure communication channels without compromising on performance.

Fast generation of RSA keys using smooth integers was explored in a study [18] by Dimitrov, Vassil, Luigi Vigneri, and Vidal Attias, presenting a novel method for accelerating RSA key generation and enhancing the performance of cryptographic systems. This method simplifies the key generation process, making RSA encryption more accessible and efficient for a wider range of applications.

A computational study [19] focused on asymmetric cryptography, emphasizing its importance in safeguarding digital communications, and highlighting the role of confidentiality during information exchange. This study reaffirms the critical role of asymmetric cryptography in protecting data privacy, reinforcing the need for robust encryption methods in today's interconnected world.

In the study by Al\_Barazanchi, Israa, et al., a modified RSA-based algorithm offering double security was developed, demonstrating an innovative approach to strengthening RSA encryption and providing a more secure option for protecting sensitive data [20]. This modification enhances the security features of RSA, making it a more formidable barrier against cyber threats.

RSA acceleration utilizing parallelization was proposed in the study [21], showcasing how parallel processing can significantly enhance the efficiency of RSA encryption and decryption processes. This innovation leverages computational resources to streamline cryptographic operations, reducing latency in secure communications.

An improvement to the RSA algorithm using Euclidean techniques was presented by authors Lizy .et .al, enhancing its efficiency and contributing to ongoing efforts to refine cryptographic techniques [22]. By incorporating mathematical optimizations, this work advances the state-of-the-art in RSA encryption, offering a more streamlined approach to data security.

# Proposed Methodology

This section discusses the proposed cryptographic algorithm, an improved variation of cryptography algorithm that combines two levels of encryption to enhance data security. The first layer is ASCON, an effective light-weight cryptographic algorithm and the second layer consists of a Dynamic RSA algorithm, which has an improved run-time compared to the traditional RSA. This Hybrid Cryptographic Algorithm offers both symmetric and asymmetric cryptography features, which improves the data security and maintains confidentiality. The following subsections describe this algorithm.

3.1. Description

In the proposed technique, a sophisticated amalgamation of two encryption layers fortifies data security. The primary layer harnesses the formidable ASCON Encryption, renowned for its lightweight attributes and robust symmetric-key foundation shown in Figure 1. This layer employs a structural framework blending elements of substitution/permutation, drawing inspiration from the renowned Feistel Structure, and enriched by the theoretical underpinnings of Shannon's cryptography. The infusion of logical operations such as XOR, AND, OR, shifting, and swapping further enhances the encryption's resilience against cryptographic attacks.

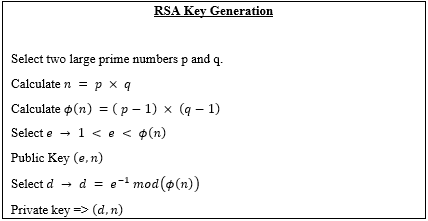
The second layer, anchored by Dynamic RSA, embraces an asymmetric cryptographic paradigm. This facet empowers the encryption process with dynamic computational capabilities, allowing specific operations on encrypted data to yield encoded results akin to the product of equivalent operations on plaintext. This bi-layered approach bestows a dual-tiered security mantle upon the plaintext, ensuring robust protection. To unveil the original content, decryption follows a reverse sequence, systematically reversing the encryption layers applied to the ciphertext.

3.2. Key Generation Process

The key is the critical component of the algorithm in the process of encryption and decryption. The diffusion and confusion techniques used to generate the proposed algorithm strengthen the key. The strength of key generation results in increased security, better encryption complexity, and decreased knowledge of the key by attackers.

The key for ASCON is generated by using the help of random library in python. Firstly, A random number is generated using this library and a 128-bit key is generated by using a random function. In Addition, We generate a 128-bit nonce before the encryption process takes place.

In RSA key generation, two large prime numbers are chosen randomly. Their product forms the modulus of the RSA key pair. The Euler's totient function is computed, and a public exponent is selected, typically a small prime number. The public key is formed with the modulus and the public exponent. Finally, the private exponent is calculated as the modular multiplicative inverse of the public exponent. The public key is shared openly, while the private key is kept secret, ensuring secure communication and data integrity. The key generation process is depicted in Algorithm 1.



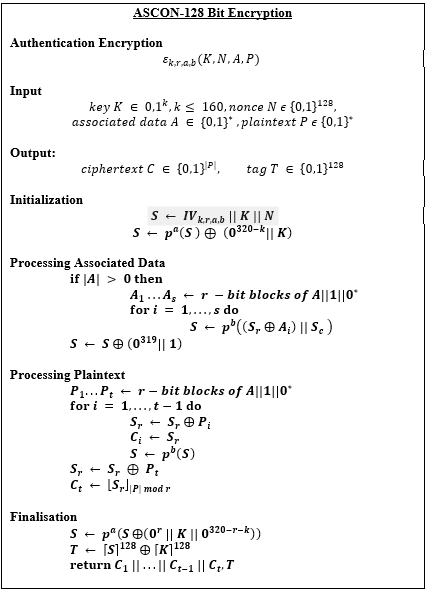
Algorithm 1: RSA Key generation process

3.3. Encryption

In the proposed algorithm, ASCON-128 bit is used for the first level encryption and Dynamic RSA for the second level encryption as the encryption process unfolds in two stages.

3.3.1. First Level Encryption (ASCON-128 bit Encryption):

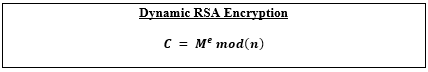
The plaintext message is first encoded using ASCON-128 bit encryption algorithm as shown in Algorithm 2. ASCON encrypts the plaintext message, generating ciphertext. The resulting ciphertext serves as the input for the next encryption stage.



Algorithm 2: ASCON Encryption

3.3.2. Second Level Encryption (Dynamic RSA Encryption):

The ciphertext from the first level encryption is encrypted using Dynamic RSA encryption. RSA's public key, which was dynamically generated for this session, is used for encryption. The RSA encryption process transforms the ASCON encrypted ciphertext into a 2-level encrypted ciphertext, ensuring additional security to the data. The algorithm for the same is depicted in Algorithm 3.



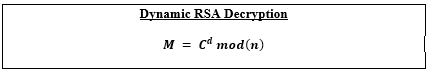
Algorithm 3: Dynamic RSA Encryption

3.4 Decryption

Similar to the encryption process, the decryption process consists of two stages. First step being decryption using Dynamic RSA Decryption and then by ASCON Decryption.

3.4.1. Second Level Decryption (Dynamic RSA Decryption):

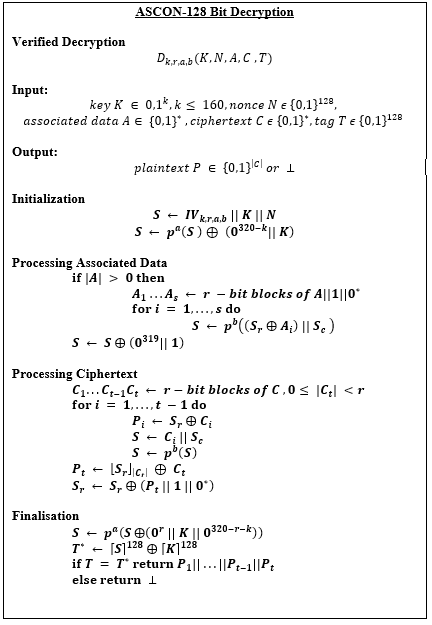
The ciphertext is first decrypted using Dynamic RSA decryption as indicated in Algorithm 4. RSA's private key, which corresponds to the public key used for encryption, is used for decryption. The RSA decryption process transforms the encrypted ciphertext back into the derived plaintext, recovering the data encrypted at the second level.



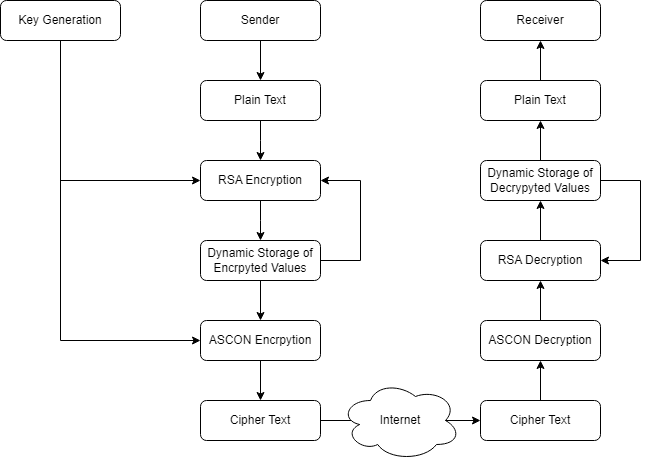
Algorithm 4: Dynamic RSA Decryption

3.4.2. First Level Decryption (ASCON-128 bit Decryption):

The derived plaintext from the second level decryption (resulting from RSA decryption) is decrypted using ASCON-128 bit decryption algorithm. ASCON decrypts the derived plaintext, recovering the original plaintext message encrypted at the first level. The algorithm for this process is shown in Algorithm 5.



Algorithm 5: ASCON Decryption



**Figure 1** Block diagram of our proposed hybrid algorithm

# Implementation and Simulation

The proposed algorithm, aimed at improving data security in data transfer applications, was simulated, and implemented on a Lenovo IdeaPad L340 laptop. Equipped with an Intel Core i5-9300H processor, the laptop provided robust computational power for intensive algorithmic processing. Complementing the processor, the inclusion of a GTX1650 GPU accelerated graphics processing tasks, crucial for image rendering and analysis.

The laptop's 8GB of RAM allowed seamless transitions between algorithm development, testing, and evaluation tasks. Utilizing Python 3.9 within Jupyter Notebook, developers harnessed the versatility of Python's scientific computing libraries, such as NumPy for code execution. The interactive environment of Jupyter Notebook provided a user-friendly interface for iterative algorithm refinement and experimentation.

# Results and Discussions

Several experiments are performed to verify the proposed algorithm’s quality, compared to other lightweight encryption algorithms and current symmetric and asymmetrically key encryption techniques such as A.E.S., D.E.S. and Blowfish R.S.A., E.G.A.M.A.L. The analysis of the algorithms is given below. The experiment was conducted with different file sizes in kilobytes (K.B.) as well as time estimates for encryption and decryption (seconds).

5.1. Time Complexity

With a 128-bit key size, the attacker would need to find 2^128 possible keys. As a result, the time complexity of 2^128 for obtaining the proper key is O on average (1). In reality, the proposed algorithm has a time complexity comparable to AES, but it is more efficient since there are no more repetitions than A.E.S. and the rest of the similar algorithms.

5.2. Execution Time

One of the most crucial aspects to consider while building cryptography is execution time. The overall time taken to encrypt/decrypt the unique data is known as the encryption cryptographic execution time. This cryptographic algorithm was implemented in Python and Jupyter Notebook.

Execution time is the algorithm's overall time to complete the execution and the processes. Execution time has two times- encryption and decryption time. Encryption time is the overall time the code takes to convert the original message into the cipher text. Decryption time is the time the algorithm takes to convert the cypher text back into the message, .i.e, in a readable format. We have compared our improved rsa algorithm with the El Gamal encryption algorithm based on execution time. Our improved rsa algorithm is also compared based on encryption and decryption times with the original rsa algorithm. We have also compared a symmetric algorithm AES based on execution time.

The evaluation test has been done in different sizes of text files. The experiment was performed 5 times on each file size and Table 1 shows the average time to execute the encryption and decryption process for text files. The table indicates the execution time in milliseconds of equivalent algorithms with various file sizes. It is obvious that the proposed algorithm takes less time than the existing N.E.L.C algorithm as depicted in Figure 2.

5.3. Throughput

The throughput rate can be used to evaluate the algorithm’s effectiveness. The algorithm’s throughput is directly related to its performance; the higher the performance/the higher the throughput. The formula for calculating the transfer rate of encoder technology is as follows:

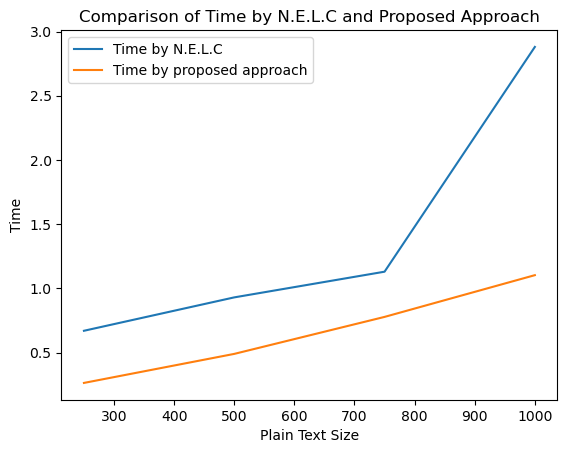
Throughput = Plain Text Size/Encoding Time.

The results of throughput is shown in Table 1. The throughput comparison of the proposed algorithm with N.E.L.C hybrid algorithm is shown in Figure 3.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Plaintext Size (kB) | Enc/Sec Time (S) | Thooughput (Kb/Sec) | Delay  (S) | Latency  (s) |
| 255 kb | 0.32 | 767.34 | 0.35 | 0.01 |
| 500kb | 0.68 | 734.43 | 0.72 | 0.03 |
| 750kb | 1.00 | 747.01 | 1.08 | 0.06 |
| 1 MB | 1.33 | 749.06 | 1.47 | 0.09 |
| Average | 0.83 | 749.46 | 0.905 | 0.04 |

**Table 1**

Computation Time and Throughput Analysis of proposed algorithm



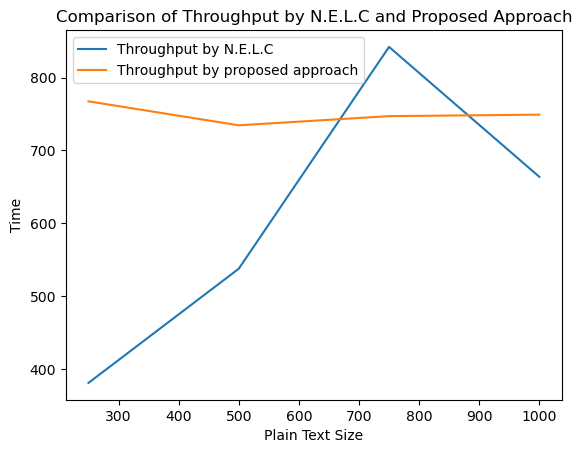
**Figure 2** Execution Time Comparison with N.E.L.C algorithm

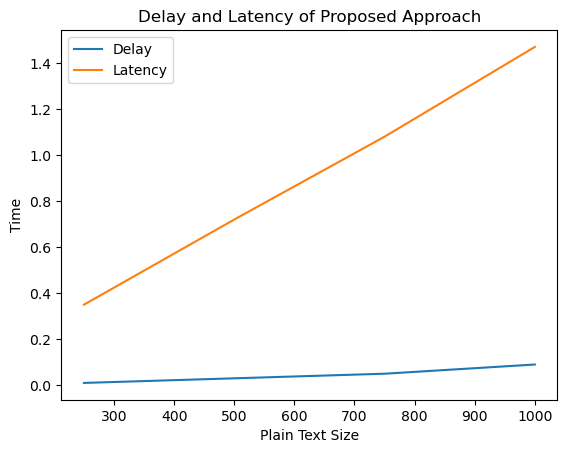
5.4. Delay and Latency

In cryptography, delay and latency play crucial roles in determining the efficiency of an encryption algorithm. Delay refers to the time taken for a message to travel from the sender to the receiver, while latency represents the time delay incurred during the processing of the message.

The proposed algorithm's delay and latency characteristics were evaluated through comprehensive testing. Multiple experiments were conducted, varying the input sizes and types of data. Each experiment was repeated five times to ensure statistical reliability.

Table 1 presents the average delay and latency values obtained from the experimental tests. It illustrates the delay in message transmission and the latency introduced during the encryption and decryption processes across different file sizes. Figure 4 graphically represents the Delay and Latency of the proposed algorithm.

**Figure 3** Throughput Comparison with N.E.L.C algorithm



**Figure 4** Delay and Latency of the proposed approach

# Comparative Analysis

In this section, comparative studies are conducted to present and verify the proposed algorithm’s feasibility and describe the analysis of the difference between the proposed algorithm, namely with a hybrid, light-weight, symmetric and asymmetric encryption algorithm that is commonly used.

The comparative studies divided into two sections describe the following.

*6.1. Comparative study of some symmetric and asymmetric algorithm*

The proposed algorithm was compared to several symmetric and asymmetric encryption algorithms frequently used for security information in cloud computing in the second section. The comparative study based on evaluated parameters commonly used for evaluated the Enc/Dec processes, such as Structure, Key size, Block Size, Possible Key, Execution Time, Cipher Type and Security Power parameters, shown in Table 2.

A faster algorithm requires fewer computational resources, such as memory. By reducing the execution time of an algorithm, the cost required in terms of resources can be reduced. In the modern era, cloud computing platforms such as amazon web service (AWS) employ virtualization techniques to consolidate physical resources into virtual instances. The cost of using cloud resources is calculated in terms of time. Hence by reducing the execution time, usage of cloud resources is minimized, leading to cost saving in terms of hardware requirement.

The algorithm is often used in real-time applications where speed is crucial. Reducing the time required for an algorithm can improve operational efficiency and reduce associated costs. As the size of the input data increases, algorithms with higher time complexity can become prohibitively expensive to execute. By reducing the time complexity, an algorithm can scale more effectively.

**Table 2**

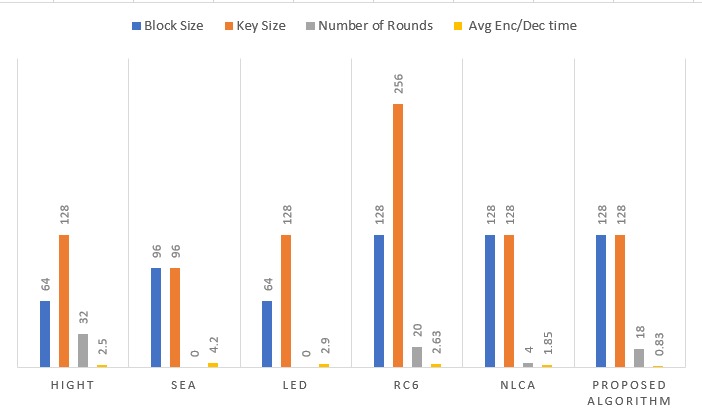
Comparative study of some of the symmetric and asymmetric algorithm.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | AES [28] | BLOWFISH [29] | *SIT* [30] | HOMOMORPHIC  RSA [31] | HOMOMORPHIC ELGAMAL [31] | PROPOSED ALGORITHM |
| Structure | Su-lbs-Per | Feistel | Feistel/SP | modular exponentiation | modular exponentiation | SP/modular exponentiation |
| Algorithm | symmetric | symmetric | symmetric | asymmetric | asymmetric | symmetric/asymmetric |
| Key Size | 128 bit | 64 bits | 64 | Key Bit Random | Key Bit Random | 128 |
| Block Size  Key space analysis | 128, 192, 256 bits  2128, 2192 Or 2256 bits | 32–448 bits  232–2448 bits | 64  264 | 512/1024 Random | 512/1024 Random | 128  2128 bits |
| Deposit Of Keys | needed | needed | needed | no | no | needed |
| No. Of Round | 10, 12, 14 | 16 | 5 | Random | Random | 18 |
| Encryption Process | Moderate | Moderate | Faster | faster | Moderate | Faster |
| Decryption Process | Moderate | Moderate | Moderate | faster | Moderate | Faster |
| Power Consumption | Low | Low | Moderate | high | high | Moderate |
| Security | Secure | Secure | Secure | Secure | Secure | High Secure |

**Table 3**

Comparative in terms of flexibility, architecture, security and limitations

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Algorithm | HIGHT [23] | SEA [24] | LED [25] | RC6 [26] | NLCA [27] | Proposed Algorithm |
| Structure | Feistel | Feistel | Feistel | Feistel | Feistel + SP | SP |
| Layer | 1 | 1 | 1 | 1 | 1 | 2 |
| Block size | 64 bits | 48, 96, 144 bit | 64 or 128 | 128 bits | 128, or 256 | 128 |
| Key size | 128 bits | 48, 96, 144 bit | 64 or 128 | 128, 192, 256 bits | 1,28,256 | 128 |
| No. of Round Possible key | 32 256 bits | Variable | Variable | 20 | 4 | 18 |
| 248, 296 Or 2144 bits | 264,2128 bits | 2128,2196 bits | 2128,2256 bits | 2128, |
| Average time (S) | 2.4 | 4.2 | 2.9 | 2.63 | 1.89 | 0.8364 |
| Mathematical Operations Used | Addition, subtraction, XOR, Shifting. (8 bits) | XOR, rotations, 2n mod addition, substitution (8 bits) | XOR, rotations,  2n mod addition, substitution (6 bits) | Addition (2’s comp).  Variable Rotation, XOR, (16 bits) | Shifting,  Substitution (4 bits)  XOR, XNOR, | Shifting, Substitution (16 bits), XOR |
| Security rate | Secure | Secure | Secure | Secure | Secure | Highly Secure |



**Figure 5** Comparative analysis with other light weight techniques

*6.2. Comparative analysis of the proposed algorithm with a light-weight hybrid cryptographic algorithm*

The first section was conducted, taking into account some evaluated parameters commonly used for evaluated the encoding and decoding processes, such as Block Size, execution time, Key Length, Possible Key, Mathematical Operations, Cipher Type and Security Power parameters, as shown in Table 3 and Figure 5.

# Security Analysis and C.I.A Achievements

The strength of a cryptographic algorithm depends on its resilience to various forms of attack, as the attacker’s endeavor to compromise the confidentiality of information through multiple strategies. Cryptographic processes must be designed to withstand all the potential attacks to ensure the sustained security of sensitive information.

*7.1. Brute force attack*

The proposed algorithm demonstrates strong defense capabilities against brute force attacks owing to the combination of ASCON 128-bit and RSA encryption techniques. ASCON 128-bit encryption, with its 128-bit key size, creates a formidable barrier against brute force attacks by offering an extensive key space that renders exhaustive search impractical. Similarly, RSA encryption, leveraging large key sizes ranging from 2048 to 4096 bits, presents a formidable challenge to attackers attempting to factor the modulus.

By employing a hybrid model that integrates both ASCON-128 bit and RSA encryption, the algorithm further enhances its resilience against brute force attacks. Even if one layer of encryption were compromised, the other layer would remain intact, significantly increasing the computational effort required for attackers to succeed. This multi-layered defense strategy ensures that the encrypted data remains secure and protected against brute force attacks, bolstering the overall security posture of the algorithm.

**Table 4**

|  |  |  |
| --- | --- | --- |
| Digits | No of operations | Time (hours) |
| 50 | 1.4 × 1010 | 0.0004 |
| 75 | 9.0 × 1012 | 0.28 |
| 100 | 2.3 × 1015 | 74 |
| 200 | 1.2 × 1023 | 3.8 × 109 |
| 300 | 1.5 × 1029 | 4.9 × 1015 |
| 500 | 1.3 × 1039 | 4.2 × 1025 |

Size of key vs Time taken to crack it

Cracking RSA encryption involves factoring the RSA modulus (n), the product of two large prime numbers, which is mathematically complex and computationally intensive. The Schroeppel algorithm, referenced by the authors of RSA in 1978, estimates factoring time based on the size of n. Assuming one microsecond per operation, factoring 50-digit n would take around 0.0004 hours, while 100-digit n requires 74 hours, and 500-digit n necessitates 4.2 septillion hours. These time estimates illustrate how the duration required to factor n rises rapidly with the size of the RSA modulus shown in Table 4.

*7.2. C.I.A. Achievement*

• Confidentiality

This indicates that unauthorized individuals are involved when information is shared. The suggested solution establishes confidentiality by encrypting all transmitted entities and parameters.

• Data integrity

Data integrity ensures that no modifications are made to the user as a result of insertion, deletion, or alteration. In other words, although the data has been manipulated, the receiver should offer specific processes to ensure that new information is obtained. Data integrity was achieved in this approach via segmentation.

• Availability

This means to ensure availability and access knowledge to users wherever the need arises. The suggested cryptographic algorithms are hypothetical techniques and are therefore useable at all times. It also supports text formats for broadband data encryption and decryption without loss of information, without losing any scheme where not lost any bit during transmission. The scheme also tests the proposed algorithm encryption using the white spaces and special characters sizes up to 10,000 plain text character.

# Conclusion

Amidst the growing popularity of remote connectivity and data storage solutions, there is a pressing need for enhanced data security measures. This research introduces a novel Lightweight Hybrid Cryptographic Algorithm, leveraging a two-layer encryption approach for heightened data secrecy. The first layer employs a unique 128-bit lightweight cryptography technique, integrating Feistel and permutation/substitution processes with Shannon's diffusion/confusion theory to bolster encryption complexity. The second layer utilizes the dynamic property of the RSA algorithm to enhance data security and transfer speeds.

Our dynamic approach stores the encryption and decryption values of characters, reusing these values for repeated characters to reduce computational complexity. Compared to existing strategies, including classical RSA, El Gamal, and AES, our approach demonstrates superior encryption and decryption times, reduced energy consumption, and lower costs.

Experimental validation reveals that our algorithm outperforms other lightweight (HIGHT, SEA, LED, RC6, NLCA) and symmetric/asymmetric (AES, Blowfish, SIT, Homomorphic RSA, Homomorphic ElGamal) encryption algorithms in encryption time, throughput, and security level. Rigorous testing against common cryptographic attacks confirms the algorithm's robustness, ensuring confidentiality, integrity, and availability.

The proposed technique holds promise for future applications, potentially yielding significantly better outcomes in various data transfer scenarios and enhancing overall security.

REFERENCES

[1] Thabit, Fursan, et al. "A Novel Effective Lightweight Homomorphic Cryptographic Algorithm for data security in cloud computing." *International Journal of intelligent networks* 3 (2022): 16-30.

[2] Krishnadoss, Pradeep, et al. "Dynamic Approach for Time Reduction in RSA Algorithm through Adaptive Data Encryption and Decryption." *International Journal of Intelligent Engineering & Systems* 17.4 (2024).

[3] Hatzivasilis, George, et al. "A review of lightweight block ciphers." Journal of cryptographic Engineering 8 (2018): 141-184.

[4] Thabit, Fursan, Sharaf Alhomdy, and Sudhir Jagtap. "A new data security algorithm for the cloud computing based on genetics techniques and logical-mathematical functions." *International Journal of Intelligent Networks* 2 (2021): 18-33.

[5] Mohamad, Mohd Saiful Adli, Roshidi Din, and Jasmin Ilyani Ahmad. "Research trends review on RSA scheme of asymmetric cryptography techniques." Bulletin of Electrical Engineering and Informatics 10.1 (2021): 487-492.

[6] Munjal, Kundan, and Rekha Bhatia. "A systematic review of homomorphic encryption and its contributions in healthcare industry." Complex & Intelligent Systems 9.4 (2023): 3759-3786.

[7] Kapoor, Vivek, and Rahul Yadav. "A hybrid cryptography technique for improving network security." International Journal of Computer Applications 141.11 (2016): 25-30.

[8] C. A. Lara-Nino, A. Diaz-Perez and M. Morales-Sandoval, "Elliptic Curve Lightweight Cryptography: A Survey," in IEEE Access, vol. 6, pp. 72514-72550, 2018, doi: 10.1109/ACCESS.2018.2881444.

[9] AbdElminaam, Diaa Salama. "Improving the security of cloud computing by building new hybrid cryptography algorithms." International Journal of Electronics and Information Engineering 8.1 (2018): 40-48.

[10] Hoobi, Mayes M. "Efficient hybrid cryptography algorithm." Journal of Southwest Jiaotong University 55.3 (2020).

[11] Kalpana, Parsi, and Sudha Singaraju. "Data security in cloud computing using RSA algorithm." International Journal of research in computer and communication technology, IJRCCT, ISSN (2012): 2278-5841.

[12] Mohammed, Saja J., and Dujan B. Taha. "Performance evaluation of RSA, ElGamal, and paillier partial homomorphic encryption algorithms." *2022 International Conference on Computer Science and Software Engineering (CSASE)*. IEEE, 2022.

[13] Mubeen, Saad, Elena Lisova, and Aneta Vulgarakis Feljan. "Timing predictability and security in safety-critical industrial cyber-physical systems: A position paper." *Applied Sciences* 10.9 (2020): 3125.

[14] Sharma, Ruchita, and Swarnalata Bollavarapu. "Data security using compression and cryptography techniques." International Journal of Computer Applications 117.14 (2015).

[15] Somani, Nikita, and Dharmendra Mangal. "An improved RSA cryptographic system." International Journal of Computer Applications 105.16 (2014).

[16] Dobraunig, Christoph, et al. "Ascon v1. 2: Lightweight authenticated encryption and hashing." *Journal of Cryptology* 34 (2021): 1-42.

[17] Daemen, Joan, et al. "Xoodyak, a lightweight cryptographic scheme." (2020).

[18] Dimitrov, Vassil, Luigi Vigneri, and Vidal Attias. "Fast generation of RSA keys using smooth integers." *IEEE Transactions on Computers* 71.7 (2021): 1575-1585.

[19] Khan, Abdul Ghaffar, Sana Basharat, and Muhammad Usama Riaz. "Analysis of asymmetric cryptography in information security based on computational study to ensure confidentiality during information exchange." *Int. J. Sci. Eng. Res* 9.11 (2018): 992-999.

[20] Al\_Barazanchi, Israa, et al. "Modified RSA-based algorithm: A double secure approach." TELKOMNIKA (Telecommunication Computing Electronics and Control) 17.6 (2019): 2818-2825.

[21] Liu, Jun-Jie, Kang-Too Tsang, and Yu-Hui Deng. "A variant RSA acceleration with parallelisation." *International Journal of Parallel, Emergent and Distributed Systems* 37.3 (2022): 318-332.

[22] Lizy, R. Felista Sugirtha. "Improvement of RSA algorithm using euclidean technique." *Turkish Journal of Computer and Mathematics Education (TURCOMAT)* 12.3 (2021): 4694-4700.

[23] Hong, Deukjo, et al. "HIGHT: A new block cipher suitable for low-resource device." *Cryptographic Hardware and Embedded Systems-CHES 2006: 8th International Workshop, Yokohama, Japan, October 10-13, 2006. Proceedings 8*. Springer Berlin Heidelberg, 2006.

[24] Huang, Shih-I., and Shiuhpyng Shieh. "SEA: Secure Encrypted-Data Aggregation in Mobile Wireless Sensor Networks." *2007 International Conference on Computational Intelligence and Security (CIS 2007)*. IEEE, 2007.

[25] Bansod, Gaurav, Nishchal Raval, and Narayan Pisharoty. "Implementation of a new lightweight encryption design for embedded security." *IEEE Transactions on Information Forensics and Security* 10.1 (2014): 142-151.

[26] Rivest, R., et al. "The RC6 Block Cipher. First Adv. Encryption…(1998)."

[27] Thabit, Fursan, et al. "Exploration of security challenges in cloud computing: Issues, threats, and attacks with their alleviating techniques." *Journal of Information and Computational Science* 12.10 (2020).

[28] Wright, Marie A. "The advanced encryption standard." *Network Security* 2001.10 (2001): 11-13.

[29] M.N. Valmik, P.V.K. Kshirsagar, Blowfish algorithm, IOSR J. Comput. Eng. (2014), <https://doi.org/10.9790/0661-162108083>.

[30] Usman, Muhammad, et al. "SIT: a lightweight encryption algorithm for secure internet of things." *arXiv preprint arXiv:1704.08688* (2017).

[31] Jabbar, Ihsan, and Saad Najim. "Using fully homomorphic encryption to secure cloud computing." *Internet of things and cloud computing* 4.2 (2016): 13-18.