International Journal of Multidisciplinary Engineering in Current Research - IJMEC

Volume 11, Issue 4s, April-2026, http://ijmec.com/, ISSN: 2456-4265

Full Length Article

Enhanced Key Exchange And Lightweight Encryption For IOT Security Using Timestamp-Based OTP And Sit

Mohammed Abdul Sameer[1], Mohammed Irfan[2], Haroon[3], Mrs. Imreena Ali[4]

[1,2,3]B.E. Students; Department Of Computer Science And Engineering, ISL Engineering College, Hyderabad, India.

[4]Assistant Professor, Department of CSE ISL Engineering College, Hyderabad, India.

masameercme@gmail.com ,mi8598106@gmail.com ,haroonsaif1292004@gmail.com

Accepted 25-04-2026

ABSTRACT

In today's digital era, secure data transmission between a data owner and users is critical. This paper presents a novel framework that combines One-Time Password (OTP) generation with an RSA-based key exchange mechanism to ensure robust data protection. Each time the data owner initiates data sharing; a unique OTP is dynamically generated and encrypted using the RSA algorithm. This OTP acts as a secure access key, ensuring that only authorized users can decrypt and retrieve the shared information. The RSA encryption not only secures the OTP but also establishes a secure communication channel between the data owner and the user, preventing unauthorized access or interception during transmission. By integrating dynamic OTPs with asymmetric encryption, this scheme enhances the confidentiality and integrity of data in environments where security.

Keywords— *One-Time Password (OTP), RSA Algorithm, Data Security, Secure Data Transmission, Asymmetric Encryption, Key Exchange Mechanism, Data Confidentiality, Data Integrity, User Authentication, Secure Communication.*

Introduction

IoT is slowly becoming an integral part of our daily life. As people use more and more intelligent devices, which include smartwatches, fitness trackers that collect personal data to smart home products like smart refrigerators, locks, fire systems, and security systems that transmit critical data around the internet. As the cost of connectivity to the internet is getting cheaper every day, and with accessibility increasing allowing more and more people to connect to the internet along with their smart devices, contributing to the growth of IoT technology. These devices are often embedded systems housing a low-power processor chip that acts as the brains of the system and is connected to a variety of sensors that collect valuable data. This data can often be critical and thus raising the question of security threats [1,2]. In order to maintain the assurance of this technology, it is vital to ensure the security of such low-power devices. These devices are part of a complex network of similar devices, exchanging lots of information over the network. Each device in the network gathers data from their corresponding sensors, ranging from a temperature sensor that collects home temperature to camera sensors that monitor traffic. These sensors generate a humongous amount of data, and this data is communicated between other devices over the internet. Thus, IoT is changing the landscape of the conventional internet to the next level by connecting everything to the internet. So, security concerns related to data.confidentiality, integrity, and authentication must be considered seriously. These devices must be able to safeguard the privacy of the user data and should not compromise the integrity of the system. Nevertheless, due to its nature of openness in terms of connectivity, IoT introduces new security challenges since it is inherently vulnerable to various threats like information leakage and unauthorized usages [3,4]. One backdrop of IoT devices is that they are prone to physical attacks resulting in exposure of data stores in the components. Also, IoT devices are most commonly connected via wireless networking, making them vulnerable to security attacks and unauthorized access, resulting in data loss, data leakage, and damage to the entire network. Encryption would solve this problem of data security for regular

https://doi.org/10.63665/IJMEC.1104s.08
ISSN: 2456-4265
IJMEC 2026

devices like computers and smartphones, but using the same cryptographic algorithms for IoT devices which are embedded systems, is questionable since the hardware architecture in these devices is more diminutive and low-powered [5,6]. Hence to address these security concerns, there is a need to use the appropriate cryptographic algorithms to maintain the integrity of the system. However, conventional cryptographic algorithms are not suitable for these smart devices since they are constrained by energy consumption, computational power, memory utilization, network bandwidth. Hence the cryptographic solutions must be appropriate for low-resource hindered devices, unlike conventional algorithms that use a lot of energy and computational power performing many rounds of encryption [7–9]. This paper presents a three-module system to meet the above requirements as efficiently and reliably as possible. The proposal is to implement various lightweight cryptographic encryption and security algorithms so that the data being transferred between these IoT devices is secure and not vulnerable to breaches. The main focus is to implement these algorithms into one shared platform for all IoT devices.

Most of the secure authentication mechanisms in recent times often adopt a time based, one-time password system [10, 11]. In such a time-based One Time Password (OTP) authentication system as used in [12], the implementation is accomplished at the application level. Two applications, a server application that is running and a client application, are used in the entire process. The server application keeps accepting connection requests, and when a connection request from a client is sent to the server, it gets accepted and connects to the server.

Literature Review

With the rapid growth of the Internet of Things (IoT), secure communication and data protection have become major challenges due to the limited computational and memory capabilities of IoT devices. Traditional cryptographic algorithms, although highly secure, often require significant processing power, memory, and energy consumption, making them unsuitable for lightweight and resource-constrained environments such as embedded systems, wireless sensor networks, and portable smart devices. As a result, researchers have focused on developing lightweight cryptographic techniques that provide strong security while minimizing computational overhead.

Shah et al. [1] conducted a comprehensive study on lightweight cryptography algorithms designed for IoT environments. The authors discussed the limitations of conventional cryptographic algorithms such as Advanced Encryption Standard (AES), Secure Hash Algorithm (SHA-256), RSA, and Elliptic Curve Cryptography (ECC) when applied to low-power and resource-constrained systems. Although these traditional methods offer high levels of security, they require considerable computational resources, processing capability, and memory, which are often unavailable in embedded devices and sensor networks. The study highlighted that modern IoT applications, including healthcare monitoring systems, wearable devices, smart homes, industrial automation, and wireless sensor networks, operate on devices with limited battery power and low processing capabilities. In such environments, implementing heavyweight cryptographic algorithms can reduce system efficiency, increase energy consumption, and introduce communication delays. Therefore, lightweight cryptography has emerged as a practical solution to address these challenges.

The authors further explained that lightweight cryptographic algorithms are specifically designed to satisfy constraints related to physical device size, memory utilization, processing speed, and energy efficiency. These algorithms aim to provide adequate security while consuming fewer computational resources. The paper analyzed several lightweight encryption techniques and compared them based on factors such as execution speed, memory requirements, energy consumption, hardware complexity, and security strength.

Additionally, the research discussed the advantages and disadvantages of various lightweight cryptographic approaches. Some algorithms demonstrated excellent speed and low power consumption but provided comparatively lower security strength, while others achieved stronger security at the cost of increased computational complexity. The study emphasized the importance of selecting an appropriate lightweight cryptographic algorithm depending on the application requirements and device capabilities.

The work by Shah et al. contributes significantly to the field of secure IoT communication by providing a detailed overview of lightweight cryptographic techniques and their applicability in modern embedded systems. The study also highlights the growing need for efficient encryption methods that can ensure confidentiality, integrity, and authentication without affecting device performance. This research serves as a foundation for developing secure and efficient encryption-based systems for IoT and other low-resource computing environments.

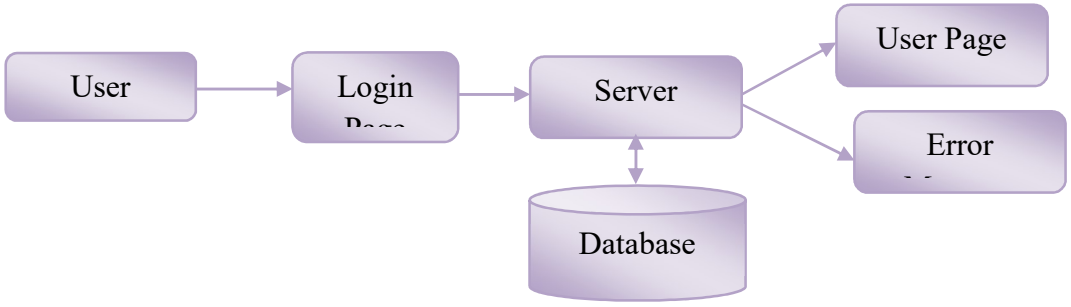
https://doi.org/10.63665/IJMEC.1104s.08
ISSN: 2456-4265
IJMEC 2026

Methodologies

The proposed system architecture consists of three basic modules as shown in Figure 1 one handles authentication, another secures the communication channel, and another ensures that data privacy is not compromised through device encryption. The authentication module uses Time-based OTP as a multi-factor authentication method in addition to a traditional ID and password. The Data Encryption module implements a lightweight symmetric key cryptography used for data encryption for both storage and transmission. The symmetric key is exchanged using a lightweight RSA algorithm suitable for IoT devices. This kind of system is designed to deal with security challenges in IoT devices, ensuring adequate security to the data at the same time reducing the computational footprint with the use of lightweight cryptography.

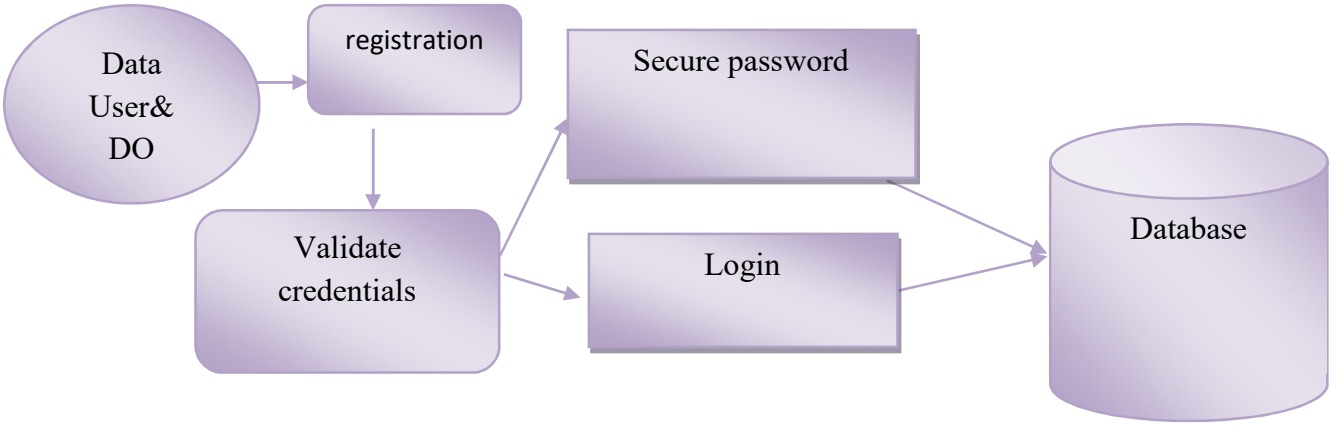
User Interface Design

To connect with server user must give their username and password then only they can able to connect the server. If the user already exits directly can login into the server else user must register their details such as username, password, Email id, City and Country into the server. Database will create the account for the entire user to maintain upload and download rate. Name will be set as user id. Logging in is usually used to enter a specific page. It will search the query and display the query.



Data User &Data Owner Registration Module

This module handles the initial registration of both users and data owner into the system. When a new user or Do signs up, the system generates a unique digital identity using the OTP algorithm, which ensures tamper-proof credentials. The identity is then recorded onto the Data owner to make it immutable and verifiable. During registration, operators are assigned special privileges to upload and manage data blocks, while users are provided access credentials to retrieve and view data securely.



Technique Used Or Algorithm Used

Proposed Algorithm
Asymmetric Encryption (RSA),Dynamic One-Time Password (OTP) Generation

- In the modern era of digital communication and information technology, the security and privacy of data have become paramount. As the internet continues to grow exponentially, individuals, organizations, and governments face increasing

https://doi.org/10.63665/IJMEC.1104s.08
ISSN: 2456-4265
IJMEC 2026

threats from cyberattacks, eavesdropping, and unauthorized access to sensitive information. To counter these threats, cryptography, the art and science of securing communication, plays a crucial role. Cryptography ensures that information can only be accessed and understood by the intended recipient while remaining unintelligible to unauthorized parties. Among the various cryptographic techniques developed, **asymmetric encryption**, particularly the RSA (Rivest–Shamir–Adleman) algorithm, has emerged as a cornerstone of modern digital security due to its robustness, versatility, and widespread adoption.

- Asymmetric encryption, also referred to as public-key cryptography, fundamentally differs from traditional symmetric encryption methods. In symmetric encryption, a single shared key is used both for encrypting and decrypting data. While symmetric encryption is efficient for processing large amounts of data, it faces inherent challenges in securely distributing the shared key. The major vulnerability arises during the key exchange process; if the key is intercepted or compromised, the entire communication becomes insecure. To overcome this limitation, asymmetric encryption was introduced. The defining feature of asymmetric encryption is the use of **two mathematically related keys**: a public key and a private key. The public key can be freely distributed and used by anyone to encrypt a message, while the private key remains confidential and is used exclusively by the recipient to decrypt the message. This key pair ensures that even if an attacker intercepts the encrypted message or the public key, they cannot decrypt the information without access to the private key.
- The RSA algorithm, named after its inventors Ron Rivest, Adi Shamir, and Leonard Adleman, is one of the earliest and most influential implementations of asymmetric encryption. Introduced in 1977, RSA marked a significant milestone in cryptography, providing a practical method for secure key exchange and digital signatures. The algorithm's security is rooted in the **mathematical difficulty of factoring large composite numbers into their prime factors**. While multiplying two large prime numbers is computationally easy, the reverse process—factorizing their product—becomes extremely challenging as the numbers increase in size. This property, known as a one-way function, forms the foundation of RSA's security, making it resistant to conventional cryptanalytic attacks.
- The RSA encryption process begins with key generation. Two large prime numbers, typically hundreds of digits long, are selected randomly and independently. These primes, usually denoted as $p$and q, are multiplied together to compute the modulus $n = p \times q$, which serves as part of both the public and private keys. Next, Euler's totient function, denoted as $\phi(n) = (p-1)(q-1)$, is calculated. A public exponent $e$is then chosen such that it is relatively prime to $\phi(n)$, meaning that $e$and $\phi(n)$share no common factors other than 1. The pair (e, n)forms the **public key**, which is shared openly and used to encrypt messages. The private key exponent $d$is derived from $e$using modular arithmetic, specifically by computing the multiplicative inverse of $e$modulo $\phi(n)$. The private key (d, n)remains confidential and is used for decryption. This key generation process ensures that messages encrypted with the public key can only be decrypted with the corresponding private key, and vice versa, which also enables digital signature verification.
- Encryption using RSA involves transforming plaintext into ciphertext using the recipient's public key. Mathematically, if a message $M$is represented as a number less than n, the ciphertext $C$is calculated as $C = M^e \bmod n$. Decryption is performed using the private key, where the original message is recovered as $M = C^d \bmod n$. This process guarantees that the encrypted data remains secure during transmission over insecure channels, such as the internet. Additionally, RSA supports **digital signatures**, a crucial feature for authentication and integrity verification. To sign a message, the sender encrypts a hash of the message with their private key, creating a signature that can be verified by anyone using the sender's public key. If the signature matches the computed hash of the received message, the recipient can be confident that the message is authentic and has not been tampered with.
- One of the significant advantages of RSA is its versatility. RSA is widely used in secure communications protocols, such as **Secure Sockets Layer (SSL) and Transport Layer Security (TLS)**, which underpin secure web browsing (HTTPS). It is also employed in email encryption systems like **Pretty Good Privacy (PGP)** and in digital certificates for identity verification. Furthermore, RSA facilitates secure key exchange in hybrid cryptosystems, where it encrypts a symmetric key used for bulk data encryption. This combination leverages the efficiency of symmetric encryption for large datasets and the security of RSA for key distribution, resulting in a practical and robust security framework.
- Despite its widespread adoption, RSA is not without limitations. The algorithm's security relies on

https://doi.org/10.63665/IJMEC.1104s.08
ISSN: 2456-4265
IJMEC 2026

the size of the key; smaller keys are susceptible to brute-force attacks, while larger keys increase computational overhead. For example, a 1024-bit key, once considered secure, is now vulnerable to modern computational capabilities, prompting the adoption of 2048-bit or 4096-bit keys. Additionally, RSA encryption and decryption are relatively slower compared to symmetric algorithms such as AES (Advanced Encryption Standard), making it less suitable for encrypting large volumes of data directly. Consequently, RSA is often used in conjunction with symmetric encryption for practical applications.

- Over the decades, research in cryptography has focused on strengthening RSA against emerging threats, including advances in quantum computing. Quantum algorithms, particularly Shor's algorithm, theoretically enable efficient factorization of large integers, posing a potential risk to RSA's security. This has prompted the cryptographic community to explore **post-quantum cryptography**, developing new algorithms resistant to quantum attacks. Nonetheless, RSA remains a foundational component of digital security, and its principles continue to influence the design of modern cryptographic systems.
- RSA's impact extends beyond technical applications; it has played a pivotal role in establishing trust in digital ecosystems. E-commerce platforms, online banking, government communications, and cloud computing services all rely on RSA-based protocols to safeguard sensitive data and ensure secure transactions. By providing both confidentiality and authenticity, RSA enables users to communicate securely, conduct financial transactions, and verify digital identities, thereby fostering trust in an increasingly interconnected world.
- In conclusion, asymmetric encryption, exemplified by the RSA algorithm, represents a profound advancement in cryptography. By leveraging the mathematical properties of prime numbers and modular arithmetic, RSA allows secure communication without the need for sharing secret keys, addressing a fundamental limitation of symmetric encryption. Its applications in secure communications, digital signatures, and hybrid cryptosystems have made it a critical tool in safeguarding digital information. While challenges such as computational efficiency and emerging quantum threats exist, RSA's legacy and continued use underscore its importance as a cornerstone of modern cryptography. As digital technology evolves, the principles of asymmetric encryption will remain essential for protecting privacy, ensuring authenticity, and maintaining trust in the digital age.

The proposed algorithm consists of three main stages: Setup, Data Sharing, and Data Access.

1. Setup Phase

Before any data sharing takes place, the user (receiver) generates an RSA key pair. This consists of a public key and a private key. The user sends their public key to the data owner, while securely storing the private key. This setup ensures that only the user can decrypt messages intended for them.

2. Data Sharing Phase

Every time the data owner wants to share data with a user, a unique One-Time Password (OTP) is generated. This OTP acts as a temporary encryption key for that particular data transaction.

- The data owner uses this OTP to encrypt the sensitive data using a symmetric encryption algorithm such as AES.
- Then, the OTP itself is encrypted using the user's RSA public key, ensuring that only the intended user can decrypt it.
- Both the encrypted data and the encrypted OTP are sent to the user.

3. Data Access Phase

Upon receiving the data, the user first decrypts the OTP using their private RSA key. Since only the user holds the private key, no unauthorized third party can access the OTP. Once the user obtains the decrypted OTP, they use it to decrypt the actual data that was encrypted by the data owner.

This two-step decryption process ensures that even if the data is intercepted during transmission, it cannot be read or misused without access to the private key and the unique OTP.

Conclusion

The proposed three-module system implements user authentication using time-based OTP, exchanges keys in unsecured communication channel using a lightweight RSA algorithm using three prime numbers, and finally, data encryption using SIT algorithm. These modules form an end-to-end secured IoT system that ensures the security of the system is never compromised using lightweight cryptographic algorithms. This end-to-end security solution can be adapted to any low-power IoT device to communicate over the internet securely. Thus, the proposed lightweight algorithms are suitable solutions to the current security challenges faced by IoT devices.

Future Enhancement

For future work, The proposed system combines multiple lightweight cryptographic solutions into one shared platform, from secure authentication to data

https://doi.org/10.63665/IJMEC.1104s.08
ISSN: 2456-4265
IJMEC 2026

encryption for transmission and storage, ensuring end-to-end security for IoT devices.

Reference

[1] H. Suo, J. Wan, C. Zou and J. Liu. Security in the internet of things: a review. In Proceedings of the International conference on computer science and electronics engineering, IEEE, 3: 648–651, 2012.
[2] A. M. MohamadAl-Aboosi, S. Kamil, S. N. H. Sheikh Abdullah and K. A. Zainol Ariffin. Lightweight Cryptography for Resource Constraint Devices: Challenges and Recommendation. In Proceedings of the 3rd International Cyber Resilience Conference(CRC), IEEE, 1–6, 2021.
[3] S. Misra, M. Maheswaran, S. Hashmi. Security challenges and approaches in internet of things. Cham: Springer International Publishing, 2017.
[4] I. K. Dutta, B. Ghosh and M. Bayoumi. Lightweight cryptography for internet of insecure things: A survey. In Proceedings of the 9th Annual Computing and Communication Workshop and Conference(CCWC), IEEE, 0475–0481, 2019.
[5] P. Shah, M. Arora and K. Adhvaryu. Lightweight Cryptography Algorithms in IoT-A Study. In Proceedings of the Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(ISMAC), IEEE, 332–336, 2020.
96 V.N.H.Kollipara et al.
[6] S. A. Kumar, T. Vealey and H. Srivastava. Security in internet of things: Challenges, solutions and future directions. In Proceedings of the 49th Hawaii International Conference on System Sciences (HICSS), IEEE, 5772–5781, 2016.
[7] N. A. Gunathilake, A. Al-Dubai and W. J. Buchana. Recent Advances and Trends in Lightweight Cryptography for IoT Security. In Proceedings of the 16th International Conference on Network and Service Management(CNSM), IEEE, 1–5, 2020.
[8] M. Katagi and S. Moriai. "Lightweight cryptography for the internet of things." Sony Corporation, 7–10, 2008.
[9] T. Eisenbarth, S. Kumar, C. Paar, A. Poschmann and L. Uhsadel. A survey of lightweight-cryptography implementations. IEEE Design & Test of Computers, 24(6):522—533, 2007.
[10] D. M'Raihi, David, S. Machani, M. Pei, and J. Rydell. Totp: Time-based one-time password algorithm, Internet Engineering Task Force, RFC: 6238, 2011.
[11] M'Raihi, David, M. Bellare, F. Hoornaert, D. Naccache, and O. Ranen. Hotp: An hmac-based one-time password algorithm. In The Internet Society, Network Working Group. RFC4226, 2005. [12] M. L. T. Uymatiao and W. E. S. Yu. Time-based OTP authentication via secure tunnel (TOAST): A mobile TOTP scheme using TLS seed exchange and encrypted offline keystore. In Proceedings of the 4th IEEEInternational Conference on Information Science and Technology, IEEE, 225–229, 2014.
[13] D. Kumar, A. Agrawal and P. Goyal. Efficiently improving the security of OTP. In Proceedings of the International Conference on Advances in Computer Engineering and Applications, IEEE, 912–915, 2015.
[14] V. L. Shivraj, M. A. Rajan, M. Singh and P. Bala Muralidhar. One time password authentication scheme based on elliptic curves for Internet of Things (IoT). In Proceedings of the 5th National Symposium on Information Technology: Towards New Smart World, IEEE, 1–6, 2015.
[15] K. S. Roy and H. K. Kalita. A survey on authentication schemes in IoT. In Proceedings of the International Conference on Information Technology(ICIT), IEEE, 202–207, 2017.
[16] M. Abd Zaid, Mustafa, and S. Hassan. Lightweight RSA Algorithm Using Three Prime Numbers. Journal of Engineering and Applied Sciences, 14(5): 9032–9035, 2019.
[17] J. Sahu, V. Singh, V. Sahu, and A. Chopra. An enhanced version of RSA to increase the security. Journal of Network Communication and Emerging Technologies, 7(4), 1–4, 2017.
[18] T. K. Goyal and V. Sahula. Lightweight security algorithm for low power IoT devices. In Proceedings of the International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, 1725–1729, 2016.
[19] V.G. Kumar Kiran, S.J. Mascarenhas, S. Kumar, J. Pais Viven Rakesh. Design and implementation of Tiny encryption algorithm. International Journal of Engineering Research and Applications, 5(6): 94–97, 2015.
[20] R. Beaulieu, D. Shors, J. Smith, S. Treatman-Clark, B. Weeks, and L. Wingers. SIMONandSPECK:BlockCiphersfortheInternet of Things. NIST Lightweight Cryptography Workshop., 1–15, 2015.
[21] M. Usman, I. Ahmed, M.I. Aslam, S. Khan, and U.A. Shah. SIT: a lightweight encryption algorithm for secure internet of things. International Journal of Advanced Computer Science and Applications, 8(1): 1–10, 2017.

https://doi.org/10.63665/IJMEC.1104s.08
ISSN: 2456-4265
IJMEC 2026