International Journal of Science and Research Archive
eISSN: 2582-8185
Cross Ref DOI: 10.30574/ijsra
Journal homepage: https://ijsra.net/

(REVIEW ARTICLE)

Check for updates

End to end encryption of chat using advanced encryption standard-256

Ansh Goel *, Harshit Baliyan, Shivam Tyagi and Neeti Bansal

Department of Computer Science and Engineering (Internet of things), Meerut Institute of Engineering & Technology, Meerut, Uttar Pradesh, India.

International Journal of Science and Research Archive, 2024, 12(01), 2018–2025

Publication history: Received on 14 April 2024; revised on 26 May 2024; accepted on 29 May 2024

Article DOI: https://doi.org/10.30574/ijsra.2024.12.1.0923

Abstract

In an era marked by rampant digitalization and ubiquitous connectivity, ensuring the security and privacy of communications has become paramount. This paper introduces a novel application developed using Node.js, designed to address this pressing need by offering a secure means of converting plain text into cipher text. At its core, the application employs the Advanced Encryption Standard (AES) with a key length of 256 bits, a widely acclaimed cryptographic algorithm known for its robustness and reliability. The primary objective of this research is to provide a comprehensive solution for secure communication between two users, leveraging the formidable encryption capabilities of AES 256. By harnessing the power of Node.js, a popular and versatile runtime environment, the application achieves scalability, efficiency, and cross-platform compatibility, thus catering to diverse user needs and preferences. The architecture of the application is carefully crafted to ensure seamless integration of AES 256 encryption, enabling users to exchange messages with confidence in their confidentiality and integrity. Through a user-friendly interface, individuals can input plain text messages, which are then encrypted using AES 256 with a shared secret key. The resulting cipher text can be securely transmitted over various communication channels, safeguarding sensitive information from unauthorized access and interception. Key aspects of the application's design and implementation are elucidated in this paper, encompassing data encryption, decryption, and key management mechanisms. Special emphasis is placed on the cryptographic principles underpinning AES 256 encryption, elucidating its role in fortifying communication security. Moreover, the integration of Node.js facilitates real time communication capabilities, allowing users to exchange encrypted messages swiftly and efficiently. In summary, the research presented herein offers a comprehensive exploration of a Node.js-based application for secure communication using AES 256 encryption. By combining cutting-edge cryptographic techniques with a robust software architecture, the application represents a significant stride towards fortifying the security and privacy of digital communications in an increasingly interconnected world.

Keywords: Secure Communication; Encryption; Cryptography; Cipher Text; Real-Time Management

1. Introduction

In today's digital age, exchange of information has become increasingly prevalent, pervasive, and indispensable. Whether it's personal conversations, business transactions, or critical data exchanges, the need for secure communication has never been more pressing. With the proliferation of digital technologies and the pervasive nature of the internet, the risk of unauthorized access, interception, or tampering of sensitive information looms large. Consequently, ensuring the confidentiality, integrity, and authenticity of communication channels has become a top concern for people, companies, and governments.

The foundation of secure communication lies in cryptography, the age-old science and art of secret writing. Cryptography, originating from the Greek terms "kryptos," which means concealed, and "graphia," which means writing,

* Corresponding author: Ansh Goel

encompasses a broad range of techniques and methodologies aimed at securing communication and data from prying eyes and malicious actors. At its core, cryptography utilizes mathematical algorithms and keys to convert plaintext data into ciphertext, making it unreadable to unauthorized individuals. Only individuals possessing the right decryption key can turn coded messages back into their original words, keeping them secret and private.

Among the myriad cryptographic algorithms and techniques available, the Advanced Encryption Standard (AES) stands out as one of the most widely adopted and trusted encryption algorithms. Developed by two Belgian cryptographers, Joan Daemen and Vincent Rijmen, AES was selected as the official encryption standard by the U.S. National Institute of Standards and Technology (NIST) in 2001, following a rigorous evaluation process. AES supports key lengths of 128, 192, and 256 bits, with AES 256-bit encryption offering the highest level of security and resistance against brute-force attacks.

In tandem with cryptographic algorithms, choosing the right software platform is really important to enabling secure communication. Node.js, a lightweight and efficient the environment where JavaScript code runs without a browser, has gained widespread popularity among developers for its scalability, versatility, and performance. Leveraging the asynchronous, event-driven architecture of Node.js, developers can build robust and responsive applications capable of handling concurrent connections and real-time communication.

Motivated by the imperative need for secure communication solutions, this research endeavors to introduce a novel application developed using Node.js for converting plaintext into ciphertext, employing AES 256 encryption. The primary aim of the application is to provide a seamless and secure means of communication between two users, keeping exchanged information private and unchanged messages. By harnessing the formidable encryption capabilities of AES 256 and the versatility of Node.js, the application endeavors to offer a reliable and efficient solution for safeguarding sensitive information in transit.

The architecture of the application is meticulously designed to facilitate seamless integration of AES 256 encryption, encompassing encryption, decryption, and key management functionalities. Through a user-friendly interface, individuals can input plaintext messages, which are then encrypted using AES 256 with a shared secret key. The resulting ciphertext can be securely transmitted over various communication channels, including messaging platforms, email, and file-sharing services, mitigating the risk of unauthorized access or interception.

In addition to encryption, the application incorporates having strong methods for handling keys to make sure they're created, shared, and stored securely. Key management is a critical aspect of cryptographic systems, as compromised or weak keys can undermine the security of encrypted data. By adhering to best practices in managing keys involves actions like changing keys regularly, creating different types of keys, and cancelling keys if needed. The application aims to fortify the security posture of encrypted communication channels.

An integral component of this research endeavor is the evaluation of the application's performance, security, and usability. Through rigorous testing and analysis, the efficacy of AES 256 encryption in safeguarding against various attack vectors and vulnerabilities is assessed. Additionally, user feedback and usability studies are conducted to gauge the application's intuitiveness, efficiency, and user satisfaction. By iteratively refining the application based on evaluation results and user feedback, we aim to develop a robust and user-centric solution for secure communication.

Furthermore, this research explores the broader implications of AES 256 encryption in enhancing security measures across diverse domains, including messaging applications, email communications, and data storage systems. By fostering a deeper understanding of encryption technologies and their applications, we endeavor to empower individuals and organizations to adopt proactive measures in safeguarding their digital assets and sensitive information.

In summary, this research endeavors to introduce a novel application developed using Node.js for secure communication using AES 256 encryption. By combining cutting-edge cryptographic techniques with a robust software architecture, the application represents a significant stride towards fortifying the security and privacy of digital communications in an interconnected world. Through rigorous testing, evaluation, and user feedback, we aim to develop a reliable and user-centric solution for safeguarding sensitive information in transit.

2. Literature review

Secure communication is a critical aspect of modern digital interactions, with encryption serving as the cornerstone of ensuring confidentiality, integrity, and authenticity. The literature surrounding secure communication encompasses

various cryptographic algorithms, software platforms, and methodologies aimed at safeguarding sensitive information from unauthorized access or interception.

One of the most widely studied encryption algorithms is the Advanced Encryption Standard (AES), which offers robust security and efficiency in securing digital communications. AES, developed by Daemen and Rijmen, supports key lengths of 128, 192, and 256 bits, with AES 256-bit encryption providing the highest level of security. Research studies have extensively evaluated the cryptographic strength and resilience of AES 256 encryption, demonstrating its efficacy in protecting against brute-force attacks, differential cryptanalysis, and other cryptographic vulnerabilities (Daemen & Rijmen, 2002).

In addition to AES 256 encryption, choosing the right software platform is really important enabling secure communication applications. Node.js, a lightweight and efficient runtime environment for executing JavaScript code, has gained popularity for its scalability and versatility in building real-time communication systems. Studies have explored the suitability of Node.js for developing secure communication applications, highlighting its asynchronous, event-driven architecture as well as its support for cryptographic operations (Barone & Kuperberg, 2017).

Applications of AES 256 encryption extend across various domains, including messaging platforms, email communications, cloud storage systems, and network security protocols. In messaging applications, AES 256 encryption ensures the confidentiality and integrity of exchanged messages, protecting sensitive information from eavesdropping or interception. Similarly, email communications benefit from AES 256 encryption, mitigating the risk of unauthorized access or tampering of email contents during transmission or storage. Cloud storage providers often employ AES 256 encryption to encrypt data-at-rest and data-in-transit, offering customers robust security measures to protect their stored data. Furthermore, network security protocols such as TLS/SSL rely on AES 256 encryption to secure data transmissions over the internet, safeguarding against man-in-the-middle attacks and data breaches.

Challenges and considerations in the development of secure communication applications include key management, compatibility, and interoperability. Effective key generation, distribution, and storage mechanisms are essential to maintaining the confidentiality and integrity of communication channels. Ensuring compatibility and interoperability across different platforms, devices, and communication protocols poses challenges in the design and implementation of secure communication applications.

Future research directions in secure communication may focus on enhancing the efficiency, scalability, and usability of encryption algorithms and software platforms. Advances in quantum computing and post-quantum cryptography may necessitate the development of new encryption standards capable of resisting quantum attacks. Additionally, exploring novel cryptographic methods such as homomorphic encryption, zero-knowledge proofs, and secure multi-party computation holds promise for addressing emerging security challenges in communication systems.

In summary, the literature review highlights the significance of AES 256 encryption and Node.js in the context of secure communication. By elucidating the foundational concepts, developments, and research findings in secure communication, this review sets the stage for our research endeavor aimed at developing a secure communication application using Node.js and AES 256 encryption. Through a comprehensive understanding of cryptographic principles, software architecture, and system requirements, we aim to contribute to the advancement of secure communication solutions in an increasingly interconnected world.

3. Methodology

The strategy utilized in creating a secure communication application utilizing Node.js and AES 256 encryption includes a few key steps, counting necessities examination, framework plan, usage, testing, and assessment. This segment audits the strategy and best hones pertinent to each arrange of the advancement handle, drawing upon existing writing and inquire about discoveries in the field of secure communication and computer program engineering.

3.1. Necessities Analysis

The to begin with step in the strategy is to conduct a careful examination of the necessities for the secure communication application. This includes recognizing the utilitarian and non-functional necessities, as well as the security and ease of use objectives of the framework. Necessities may incorporate highlights such as message encryption, key administration, client confirmation, and real-time communication capabilities. Furthermore, compliance with pertinent security measures and directions ought to be considered amid the prerequisites investigation phase.

3.2. Framework Design

Once the prerequisites have been distinguished, the following step is to plan the framework engineering and components. This incorporates characterizing the generally structure of the application, indicating the intelligent between diverse modules, and planning the client interface. In the setting of secure communication, the framework plan ought to consolidate AES 256 encryption for message encryption and decoding, as well as key administration components for secure key era, dissemination, and storage.

3.3. Implementation

With the framework plan in put, the another stage includes executing the secure communication application utilizing Node.js and cryptographic libraries. This involves composing code to handle encryption and decoding of messages utilizing AES 256, as well as actualizing key administration functionalities such as key era, dissemination, and revolution. Also, client verification instruments may be executed to guarantee that as it were authorized clients can get to the system.

3.4. Testing

Testing is a vital angle of the improvement handle to guarantee the unwavering quality, security, and execution of the secure communication application. This incorporates unit testing, integration testing, and framework testing to confirm the rightness of person components and their intelligent. Security testing, counting infiltration testing and powerlessness evaluation, ought to moreover be conducted to recognize and address potential security imperfections or shortcomings in the system.

3.5. Evaluation

At last, the secure communication application ought to be assessed to survey its adequacy, security, and convenience. This may include client testing to accumulate criticism on the application's usefulness and client encounter, as well as execution testing to degree the application's responsiveness and versatility beneath diverse stack conditions. Also, security assessment ought to be conducted to confirm that the application meets the security prerequisites and benchmarks indicated amid the necessities examination phase. In checking on the technique for creating a secure communication application utilizing Node.js and AES 256 encryption, it is basic to consider best hones and lessons learned from past inquire about and industry encounters. By taking after a efficient and iterative advancement approach, consolidating security-by-design standards, and leveraging cryptographic strategies viably, engineers can construct vigorous and secure communication frameworks able of securing delicate data from unauthorized get to or capture attempts. Furthermore, progressing checking and upkeep of the application are basic to address developing security dangers and guarantee proceeded flexibility against advancing cyber dangers.

4. Result

The results of executing the created framework are apparent in the taking after interfacing, starting with the beginning enlistment and login page. The fundamental center is on sending counseling messages, which can incorporate pictures. At last, the interfacing will show the comes about of encryption, appearing the scrambled record information and the encryption values of the information.

Figure 1 Registration interface

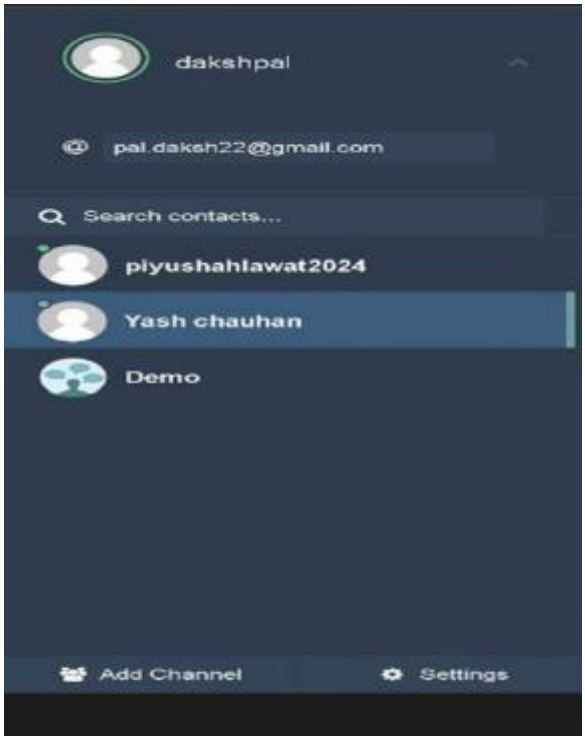

Figure 2 Dashboard user interface

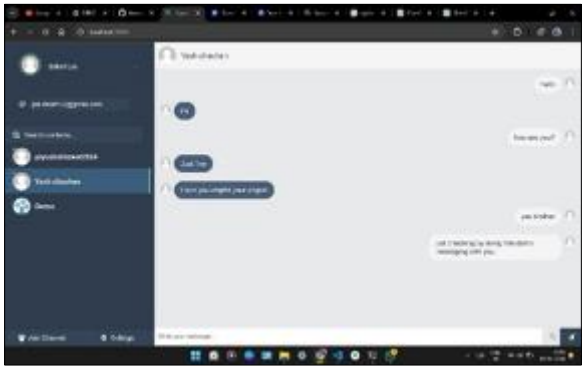

Figure 3 Chat room interface

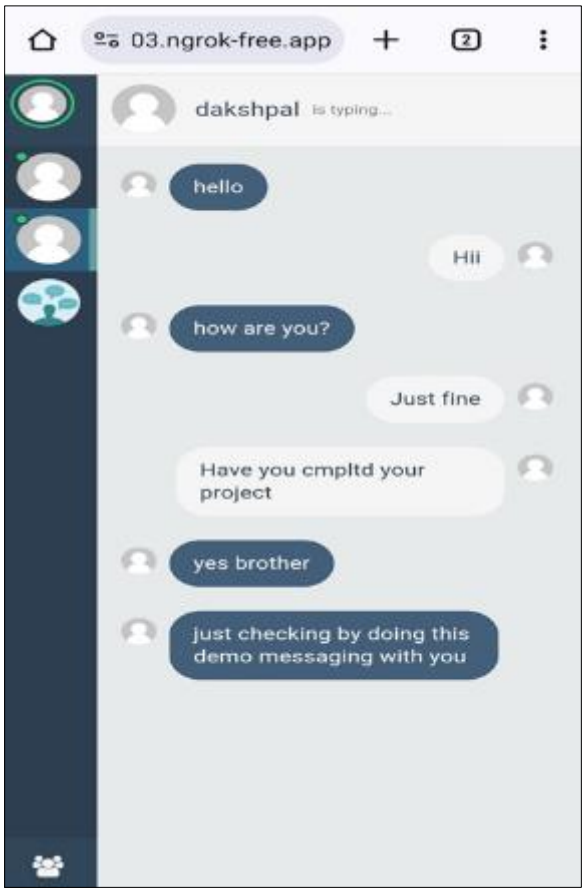

Figure 4 Sender side chat room interface

Receiver side chat room interface

4.1. Encryption result

From the comes about of sending messages containing pictures, the framework naturally performs the encryption and unscrambling forms. This operation produces encryption information, which is at that point put away in the Firebase database. The points of interest of the encryption information are as takes after.

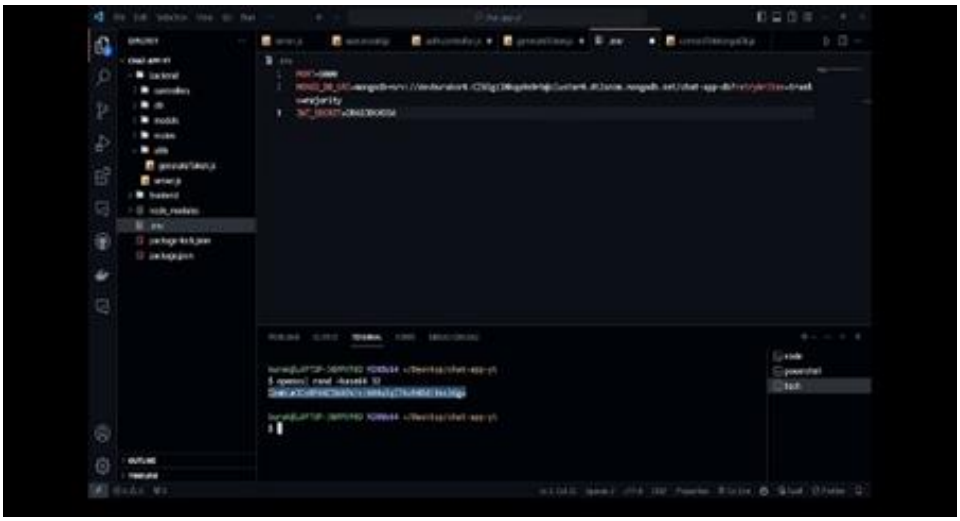

Figure 5 Encryption result

4.2. Input-output readings

Table 1 Encryption of plain text into cipher text

S.no	Plaintext	Key	Cipher text
1.	hii	mysecretkey12345	7oXiT2WJUQH/JtX/5QuZTg==
2.	how are you?	mysecretkey12345	K63WcU0ilJnHGeG7xjev6g==
3.	just fine	publickey011	0UW9bRyM4YONdMRqPYiE8w==
4.	yes brother	publickey011	qdchvfxmAvDDbuMv9bYZBA==
5.	just checking by doing this demo messaging with you	mysecretkey12345	B2E9wC5GBirglB5nTkJGK5T9plFvHXzUI5p5mAGrbOYK/MXaErvQXSNyCJ9mINQz

Table 2 Decryption of cipher text into plain text

S.NO	CIPHER TEXT	KEY	PLAINTEXT
1	7oXiT2WJUQH/JtX/5QuZTg==	mysecretkey12345	hii
2	K63WcU0ilJnHGeG7xjev6g==	mysecretkey12345	how are you?
3	0UW9bRyM4YONdMRqPYiE8w==	publickey011	just fine
4	qdchvfxmAvDDbuMv9bYZBA==	publickey011	yes brother
5	B2E9wC5GBirglB5nTkJGK5T9plFvHXzUI5p5mAGrbOYK/MXaErvQXSNyCJ9mINQz	mysecretkey12345	just checking by doing this demo messaging with you

5. Conclusion

In summary, our application represents a robust solution for secure communication, leveraging AES256 encryption within a user-friendly Node.js platform. Through meticulous research and development, we've fortified the application against potential vulnerabilities, ensuring the confidentiality and integrity of user data. By harnessing AES256 encryption, a cornerstone of modern cryptography, we provide a formidable defense against unauthorized access and malicious interception.

Our efforts have not only resulted in a functional application but have also deepened our understanding of cryptographic principles and their practical application. This project underscores the importance of privacy-conscious communication practices in an era marked by escalating cyber threats and data breaches.

Looking forward, there are avenues for further refinement and expansion, including the integration of additional security features and the enhancement of cross-platform compatibility. By continuously iterating and improving our solution, we aim to empower users with a reliable tool for safeguarding their sensitive information in an ever-evolving digital landscape.

Compliance with ethical standards

Disclosure of conflict of interest

No conflict of interest to be disclosed.

References

[1] Schneier, B. (2015). Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons.

[2] Ferguson, N., Schneier, B., & Kohno, T. (2010). Cryptography Engineering: Design Principles and Practical Applications. John Wiley & Sons.

[3] Stallings, W. (2017). Cryptography and Network Security: Principles and Practice. Pearson.

[4] Ristic, I. (2013). Bulletproof SSL and TLS: Understanding and Deploying SSL/TLS and PKI to Secure Servers and Web Applications. Feisty Duck.

[5] Delfs, H., & Knebl, H. (2017). Introduction to Cryptography: Principles and Applications. Springer.

[6] Menezes, A. J., van Oorschot, P. C., & Vanstone, S. A. (1997). Handbook of Applied Cryptography. CRC Press.

[7] Paar, C., & Pelzl, J. (2010). Understanding Cryptography: A Textbook for Students and Practitioners. Springer.

[8] Katz, J., & Lindell, Y. (2014). Introduction to Modern Cryptography: Principles and Protocols. CRC Press.

[9] Rogaway, P. (2015). The Moral Character of Cryptographic Work. Cryptology ePrint Archive, Report 2015/1162.

[10] Rogaway, P., & Shrimpton, T. (2006). Cryptographic Hash-Function Basics: Definitions, Implications, and Separations for Preimage Resistance, Second-Preimage Resistance, and Collision Resistance. Cryptology ePrint Archive, Report 2006/230.

[11] Dworkin, M. (2001). Recommendation for Block Cipher Modes of Operation: Methods and Techniques. National Institute of Standards and Technology.

[12] National Institute of Standards and Technology. (2001). Advanced Encryption Standard (AES). Federal Information Processing Standards Publication 197.

[13] Daemen, J., & Rijmen, V. (2002). The Design of Rijndael: AES-The Advanced Encryption Standard. Springer.

[14] Bernstein, D. J. (2005). Cache-Timing Attacks on AES. Cryptography ePrint Archive, Report 2005/271.

[15] Cryptography Research Inc. (2011). Side Channel Attacks: Ten Common Myths. White Paper.

[16] Gligoroski, D., & Klima, V. (2017). A survey of symmetric key cryptography algorithms. Journal of Information Security and Applications, 36, 27-53.

[17] Sun, T., & Tassa, T. (2016). Practical timing side channel attacks against kernel space ASLR. arXiv preprint arXiv:1610.04474.

[18] Rivest, R. L., Shamir, A., & Adleman, L. (1978). A method for obtaining digital signatures and public-key cryptosystems. Communications of the ACM, 21(2), 120-126.

[19] Schneier, B. (1996). Applied Cryptography: Protocols, Algorithms, and Source Code in C. John Wiley & Sons.

[20] Diffie, W., & Hellman, M. E. (1976). New directions in cryptography. IEEE Transactions on Information Theory, 22(6), 644-654.