iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** I **Month of publication:** January 2026

DOI: https://doi.org/10.22214/ijraset.2026.77146

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

Web Application for Secure Transmission of Data

Rama Subbarao Mamidipalli[1], Gangadhar Rao Mallavarapu[2], Rakesh Surya Gangadhar Mangam[3], Satya Siva Rama Krishna Bonthu[4], Rohith Sai Ramareddy Bethireddy[5], Ganga Bhavani B[6]

[1, 2, 3, 4, 5]*Department of Computer Science and Engineering, Bonam Venkata Chalamayya Engineering College, Affiliated to JNTU Kakinada, Andhra Pradesh-533210, India*

[6]*Project Guide, Department of Computer Science and Engineering, Bonam Venkata Chalamayya Engineering College, Affiliated to JNTU Kakinada, Andhra Pradesh-533210, India*

Abstract: ***In the modern digital era, the secure transmission of data over the internet in the digital world is highly required due to the rising growth in cybercrime incidents. Upon completion of the research work in the digital domain, it will be possible to develop an entire web-based application based on the concept of secure data transmission by using a combination of different hybrid techniques of cryptography based on AES, along with the RSA key exchange algorithm. With proper implementation, the strengths of both the encryption standards could be utilized while the weaknesses of the respective standards could be avoided. The application is developed with HTML and CSS for the frontend, an intuitive interface, while Python's Flask framework and MySQL are used as the server side for robust backend processing. This creates a smooth experience for the users in encrypting and also decrypting the information in a secure manner. Here, the Fernet symmetric authenticated encryption scheme from the Python Cryptography Library has been used for effective deployment of the algorithms within the given architecture. Further, the performance evaluation of the proposed model ensures the achievement of the best standards in terms of security and the level of processing efficiency, making it highly suitable for the practical implementation of secure communication within the domain. Thus, the proposed model can solve the challenges of efficiency alongside the security challenges that occur within the environment of secure communications.***

Keywords: AES algorithm, RSA algorithm, Cryptography, Data Security, Secure Communication

I. INTRODUCTION

With an exponential growth in the usage and demand of internet services and different modes of internet communication across the world, issues of data security and privacy have cropped up as two significant issues. Today, when many vital information exchanges take place through different media of communications, issues of information confidentiality, integrity, and authenticity have become extremely significant issues not just in terms of businesses and governments but individuals across the world[1]. The world of digits and data security is reaching unprecedented threat vectors of unauthorized activities like unauthorized access, mass data breach attacks, and many more. Moreover, statistics show that this cost will amount to trillions of dollars in a single year, while millions of records will be at a greater risk of exposure because of a breach in this kind of data. The alarming situation of cybercrimes, therefore, proves that secure encryption methods will require more attention to secure digital information. Apparently, traditional security balances cannot fight cyber threats effectively, thus requiring more sophisticated encryption methods. "The fundamental backbone for information security is based on cryptography, which provides various tools to safeguard information from access and misuse through the use of mathematical algorithms during transmission and storage" [2]. Amongst various tools and methods used in cryptography for information security, the encryption algorithm is the most basic and plays a very significant role in converting readable information into an unreadable format for preventing unauthorized accesses of information while in transit. Advanced Encryption Algorithm: Advanced Encryption Algorithm, also known as Advanced Encryption Standard, is considered one of the most secure encryption techniques that has less computational complexity in its application to avail maximum encryption benefits. Presently, it is being used by governments across the world as a standard encryption mechanism to store data securely in financial institutions, technological businesses, and the government itself to keep their private data. Nevertheless, there are also a few limitations that have been associated with the symmetric cryptogram encrypting component. To be a little more precise, one generalized aspect considered here is related to the safe receipt of the encrypting key such that it is not intercepted by any other "evil entity." If the key involved gets compromised in the "key exchange process" itself, the whole idea of security fails. As far as the implementation of the "RSA algorithm of safe key exchange is considered, it is clearly seen that the implementation of this algorithm is fully compliant with the "asymmetric key cryptography principle."

As opposed to the requirement of "key exchange" involved in the implementation of any "symmetric encrypting option," this option nullifies this issue of "key exchange" in a very elegant way without involving any "key sharing" between the two organizations. To outline the main aim of this study, the aim of this particular paper aims at designing an integrated web-based application through the merging of the AES and RSA algorithms in an attempt to create an even more robust algorithm for the sake of designing an even more robust encryption technique that will be capable of efficaciously leveraging the best from the combining algorithms in the designing of this particular paper. This particular paper aims to address the motivation behind the need for an important user interface through the use of an efficient encryption technique.

II. LITERATURE REVIEW

Existing studies based on secure data transmission techniques have widely evaluated several types of cryptography techniques with various advantages and disadvantages. In symmetric cryptography techniques like DES, Triple DES, or AES, data encryption techniques are widely evaluated with various advantages. The security of the data is efficiently achieved as the large amount of data can be easily encrypted with less processing time. Although the DES is a very crucial technique with historical values, it is no longer deemed a secure technique since the length of the key is too low to withstand brute force attacks. TripleDES has improved security by applying the DES algorithm three times, with the disadvantage of the processing time. AES evolved as an improved replacement for DES, resulting from a detailed selection process implemented and conducted by the National Institute of Standards and Technology (NIST). Its adoption as a federal standard represented a major leap forward in symmetric crypto technology. However, the key problem in utilizing symmetric crypto systems has remained the distribution of keys, i.e., without the interception of the encryption key during transmission. The elegance of key exchange lies in asymmetric encryption algorithms, for example, RSA; however, this is computationally expensive in handling large datasets of information. Prashant et al., in their study (2024), presented a comprehensive comparative study of AES and RSA algorithms, along with other cryptographic techniques, indicating that while AES algorithms are efficient in terms of data processing speeds, in comparison, they are 1000 times faster compared to RSA in handling larger datasets, while RSA algorithms are better in handling key transfer due to public key infrastructures, as demonstrated in Prashant et al. (2024).

Sood and Kaur undertook an extensive study of various encryption techniques like RSA, DES, AES, etc. The authors found that out of all the data encryption techniques analysed based on security strength, key management techniques, and complexity of computation involved in each method, AES has the best compromise between security and efficiency of operation in the modern world of increasing data needs and threats to data security and integrity. Singh and Supriya carried out an analysis of various stages in the evolution of encryption algorithms, beginning with historical forms of encryption up to recent computational methods, thus substantiating their claim of AES being the industry standard algorithm because of its resistance to decryption, ease of implementation in both software and hardware forms, and its versatile application in various forms [3]. The authors' findings highlighted how better security solutions are developed by a combination of both methods than either being applied in isolation to compensate for their inherent deficiencies.

Kumari and Mahato (2025) discussed the “fascinating evolution” that can be traced regarding the history of cryptography that starts from the substitution ciphers used by Julius Caesar until the present day “quantum-resistant” form of encryption that can help handle the “inherent need for increasingly robust forms of encryption that grow exponentially alongside computing power.” They identified that the “hybrid forms” that depend on more than “a single form of encryption provide far more robust security for modern applications by creating an exponentially harder problem.” The literature provided within the Globus Toolkit documentation offers a level of insight pertaining to concepts revolving around security and encryption in a distributed system, as well as the importance of secure protocols within a networked scenario [5].

It highlights that a strong level of encryption must exist within a distributed system. Though there have been extensive theoretical explorations of each of these algorithms, there has been a huge gap identified in their overall practical implementation that incorporates AES as well as RSA in an easy and interactive manner available for end users through internet applications or platforms. Moreover, most of such applications that are available are highly centered around their theoretical and practical analysis in controlled environments rather than their implementation in easier manners for users to utilize in real-time applications or scenarios. The significance of doing this research is that it is aimed to serve as a reliable platform that incorporates ease of use in association with high levels of security requirements during its overall implementation process.

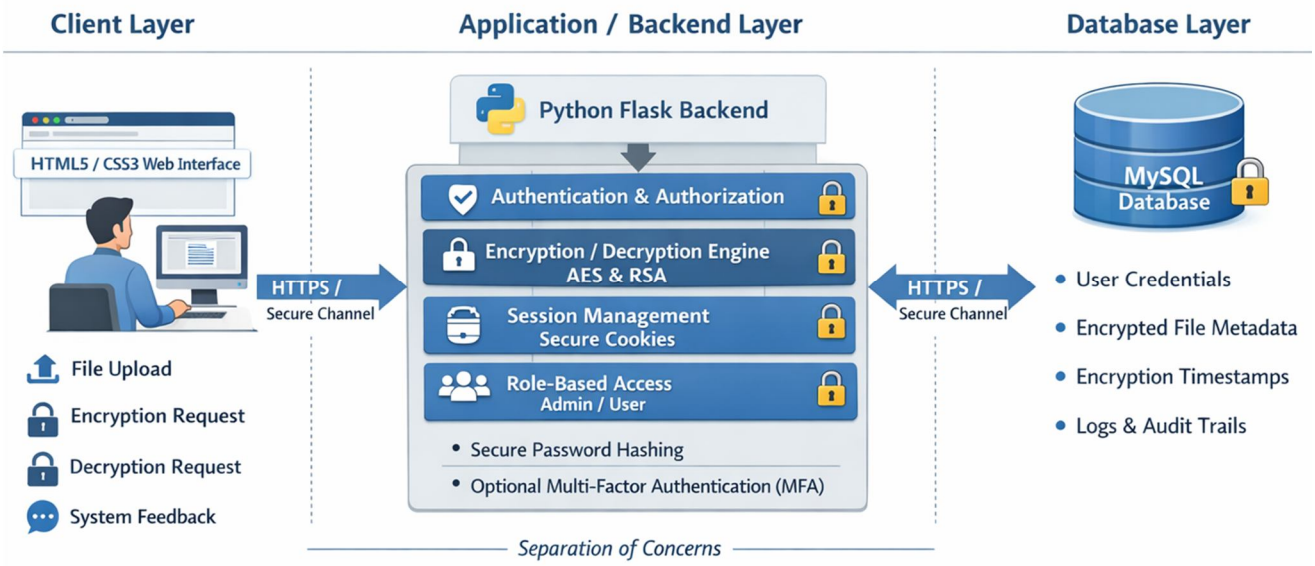
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue I Jan 2026- Available at www.ijraset.com

III. METHODS

A. *System Architecture*

The system that is proposed has a highly structured client-server architecture. Users will interact with this system through a normal web interface. They will upload files to perform corresponding encryption/decryption operations. This application is developed with front-end tools such as HTML5 and CSS3. Normal web development principles have been used to make this application highly user-friendly. Users will have a good view of system feedback during encryption/decryption operations. The backend layer employs the concept of lightweight and powerful web development technologies like Python Flask. It handles encryption operations and deals efficiently with user authentication and database interactions. It ensures clean separation of concerns. It has built-in security features.

MySQL database management will be utilized, storing the users' credentials, encrypted file information, information about the encrypted files, timestamps relating to the encryption process, and log information. The design scheme will enable maintainability, as well as optimize query performance to enable the retrieval of files quickly. User authentication techniques are used to ensure that the users who are accessing the encryption and decryption services are valid. The system utilizes secure password hashing with widely accepted standards and protocols for session handling, with the use of secure cookies for the purpose of retaining the state information of users. Role-based security is used for the administration functions.



B. *AES Algorithm Working*

AES is a symmetric block cipher that uses a block size of 128 bits, along with a key size of 128, 192, or 256 bits, to encrypt blocks of information, where the larger the key size, the more secure the information will be [1]. Here, a 256-bit key size has been used to ensure maximum security for the information stored in the database, as it can safely counter any present or future kinds of cryptanalysis attacks performed on the information. The process of AES encryption takes place through several rounds of transformation, with 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds of transformation for 256-bit keys. The round consists of four different operations:

1) *SubBytes:* Each byte of the state array in the Rijndael algorithm is replaced by another byte from a pre-defined substitution box. Adding this non-linearity makes it difficult to apply differential and linear cryptanalysis.
2) *ShiftRows:* The rows of the state matrix are shifted in a cyclical way and by different displacements, where row 0 stays unchanged, row 1 shifts left by 1 byte, row 2 by 2 bytes, and row 3 by 3 bytes. This increases dispersion across the algorithm, with a modification in a certain input having also changed several bytes in the output.
3) *MixColumns:* This operation mixes the columns using multiplication of matrices over the Galois Field GF(2^8), which results in more diffusion of data in the block, so each input bit affects many output bits. The MixColumns step is excluded from the last round.
4) *AddRoundKey:* By using XOR operations, the state is combined with one round key, which is derived from the main encryption key. This introduces the key material into each round.

The process of decryption is the inverse of the encryption process. The inverse steps involve InvSubBytes, InvShiftRows, InvMixColumns, and AddRoundKey in the opposite order. The merits of AES include faster encryption rates, better security, measured by resistance to all types of attacks, including differential cryptanalysis and linear cryptanalysis, along with easy implementation in hardware as well as software [2]. The modern processors have been equipped with the AES instruction sets that improve the rate of the encryption and decryption process.

C. RSA Algorithm Working

RSA is a public-key or "asymmetric-key" encryption method that relies upon a pair of mathematically linked keys: a public encryption key may be openly distributed, whereas a private key for decryption must be kept secret [3]. The basis upon which RSA operates is the difficulty in processing the reversal of a product of large prime numbers, a problem for which there is as yet no efficient method or algorithm implemented anywhere in the world in computer systems [1]. Some of the most critical steps in the key generation process are considered to be the selection of two large prime numbers that are often 1024 or 2048 bits long each, the computation of the product of the two prime values as the modulus value ‘n’, the computation of the value represented as $\varphi(n) = (p - 1)(q - 1)$, the selection of the public exponent ‘e’, which is necessarily higher than 65537 for efficiency considerations and is coprime with $\varphi(n)$, and the computation of the private exponent ‘d’, which is needed to satisfy the equation $(d * e) \bmod \varphi(n) = 1$.
During the encryption process, the text message sent is converted into integers. The integer form of the message is then raised to some power e modulo n: C = M^e mod n. While in the process of decrypting the data, the ciphertext that has been sent receives the value of the modular inverse of the original integer raised to some power d modulo n: M = C^d mod n. The basic formula that has been used shows that only the recipient with the matching private key will be able to decrypt any data that has been encrypted with their matching public key [4]. In this case, the RSA algorithm has an important role in key exchange security rather than data encryption. In fact, the key used for encryption by the AES algorithm has to be encrypted using the RSA key of the receiver before the information is actually sent over the channel. In this way, only the authorized party having the corresponding private key can access the actual key for decryption using the AES algorithm.

IV. RESULTS

The experimental environment was a local environment, in which a system that has 8 GB of memory, an Intel i5 processor running at 2.4 GHz, Windows11 as its operating system, along with Python 3.8 and Flask 2.0. The system tested various cases of system performance by testing files of various sizes, from 1KB of plain text up to 10 megabytes of multimedia content. Samples provided indicated that a full encryption/decryption cycle with perfect integrity was achievable. The 1 MB text file took 0.15 seconds to encrypt with AES 256. The RSA file encryption of the 256-bit AES took 0.03 seconds regardless of file size. In terms of decrypt times, this is close to the encrypt times. RSA took 0.04 seconds to decrypt, while AES took 0.15 seconds to decrypt this file. Further, the efficiency of the code is demonstrated by the performance evaluation result that the time consumed for file encryption is directly proportional to the size of the file. The code maintains consistency in efficiency with various types of files, such as text documents, images, video clips, or compressed files. When the size of the file is considered as 5MB for the pdf file encryption, the code takes a total of 0.68 seconds. When the size of the video file is considered as 10MB, the time consumed is 1.3 seconds. Further, the time consumed for the encryption of the key is constant for any file size or any file type and is considered to be 0.03 seconds. The file size, as opposed to processing time analysis, demonstrated that it is efficient to handle files of up to 10MB without any noticeable deterioration in performance or any signs of excessive use of memory. The method was seen to be far more efficient and effective as opposed to relying on using RSA for large files alone, which would have seen its processing time 50-100 times higher and also been limited owing to its maximum message size constraint. Security effectiveness testing has assured us that the encrypted files are completely unreadable without the correct decryption keys. It has been ensured that if incorrect keys are used while attempting to decrypt the files, total failure with proper error messages is achieved. Simulations of brute-force attacks have assured us that attempting such an approach to break the encryption is computationally impossible in any timely fashion. The system has been successful in preventing unauthorized access, data tampering, and replay attacks through the proper implementation of security best practices.

V. DISCUSSION

Results comprehensively demonstrate how the hybrid model of combining AES and RSA outperforms and sustains an effective, practical solution for safe data transmission in web environments. The hybrid approach has been strategically using AES's computational efficiency for encrypting large data volumes by utilizing RSA for its secure key exchange method to address the very symptom-a symmetric key distribution challenge-that has classically hindered the widespread cryptographic deployment.

The implementation of the proposed system would therefore offer an optimum balance between the twin requirements of security and performance, compared to existing encryption-based systems that use either symmetric or asymmetric methods exclusively. Systems based on RSA alone for data encryption suffer from severe performance bottlenecks when large files are involved and become practically unusable beyond a few kilobytes because of computational complexity issues. Systems based purely on AES, on the other hand, face problems of secure key distribution, which invariably forces organizations to adopt out-of-band key exchange methods. Accordingly, the advantages of employing both AES as well as RSA are numerous; in line with this fact, the encryption strength provided by 256-bit encryption in AES would effectively act as a barrier to brute-force hacking attempts, the implications of key distribution problems would be effectively handled by employing secure key transmission in RSA encryption, the performance characteristics of this system would facilitate real-time encryption of files with moderate sizes, as well as demonstrate defense in depth because of the layered encryption employed. This system would effectively provide high levels of security for data, including financial data, healthcare data, and confidential business data. There are naturally security/performance compromises to be made here. While the security of the key is very high with an RSA encryption, the computational burden is very low compared to a regular AES encryption. The system attempts to manage this in two ways: the computational burden will not be an issue whatsoever, as the RSA is used only for key encryption rather than data encryption. Naturally, there is a minor increase in memory usage because two contexts are needed: one for AES and another for RSA.

The real-time applicability of the system extends to a variety of practical scenarios, which include secure file sharing within an organization, confidential document transmission between business partners, protected cloud storage in which even the service provider does not have any access to unencrypted data, secure email attachments, healthcare data exchange as per privacy regulations, and financial transaction security. The web-based interface makes advanced cryptography accessible to users without requiring deep technical knowledge, thus promoting wider adoption of security best practices. Some future enhancements to this may include the use of digital signatures for authentication and non-repudiation of messages, inclusion of more encryption algorithms to enable algorithm agility, and integration with enterprise identity management systems. Others include key escrow mechanisms that ensure organizational data recovery, file compression before encryption to minimize costs in storage and transmission, and finally, the development of mobile applications to ensure cross-operability.

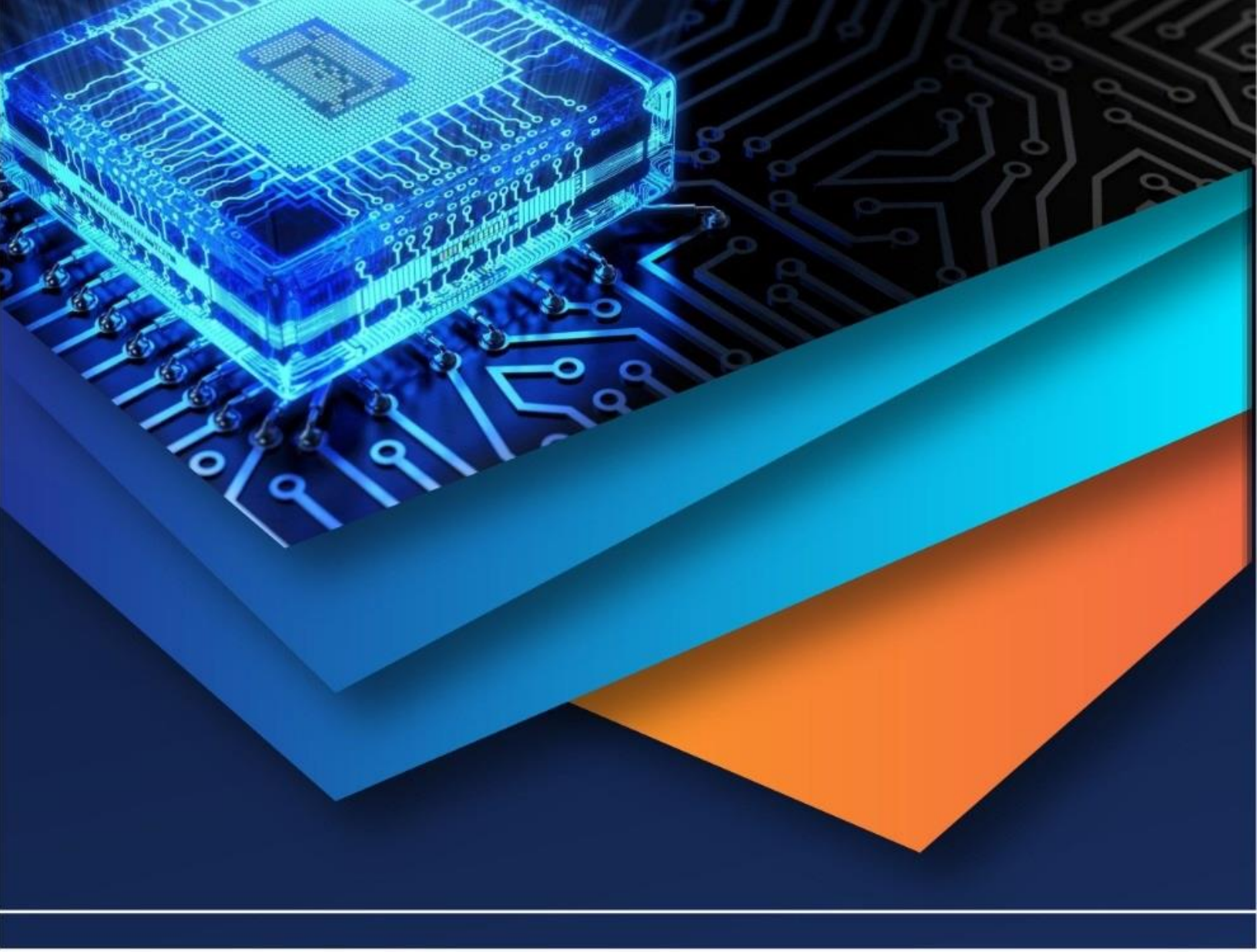
VI. CONCLUSION

Overall, the study has effectively created and evaluated a comprehensive internet application that is able to facilitate secure data transmission via a technique employing the best features of both symmetric and asymmetric data encryption. Implementation of the AES technique has proved highly successful for the efficient encryption of data via the utilization of a technique employing the features of both symmetric and asymmetric data encryption. It can be seen that the proposed system fulfils its proposed objectives with a facility that can perform user-friendly and understandable file encryption as well as decryption, even for users who are not highly proficient in matters of information security. The proposed approach also passed performance evaluations that showed that not only does it perform optimally, but it also ensures that there are strong security standards, enough to counter current cyber threats, with the two concepts covering each other well, as one method does not have what the other lacks. It successfully implemented hybrid encryption using current, open standards for encryption algorithms. The intuitive web interface is easily accessible from any modern browser. It has very efficient processing times for files up to 10 MB, with linear scalability. Key management mechanisms are secure, ensuring protection for the cryptographic keys throughout their life cycle. Authentication and authorization controls are strong, preventing unauthorized access. This system demonstrates how sophisticated cryptographic security can be made available without compromising rigor in security. Future enhancements may be the support for large files by the use of streaming encryption, the use of digital signatures for authentication and non-repudiation, more encryption algorithms to enable algorithm agility, the development of mobile applications for a wider platform support, and the implementation of advanced key management features, including key rotation and escrow mechanisms in enterprise deployments. With this, the web application demonstrates that hybrid cryptography systems show practical and deployable solutions to provide security in modern digital communications; thus, a balance between the two important and competing requirements of security strength and computational efficiency can be well achieved. The research applied therein proves that any efficient and sound cryptographic implementation could be robust for protection while remaining practical for everyday usage.

REFERENCES

[1] Prashant, Md Sohail Haque, Amrinder Kaur, Pankaj Yadav. (2024). Comparative Analysis of AES and RSA with Other Encryption Techniques for Secure Communication. International Journal of Scientific Research in Computer Science, Engineering and Information Technology.

[2] Sood, Kaur, S. A Literature Review on RSA, DES, and AES Encryption Algorithms. SCRS Publications.
[3] Singh, Supriya. A Study of Encryption Algorithms (RSA, DES, 3DES, and AES) for Information Security. International Journal of Computer Applications.
[4] Kumari, Mahato, T. K. (2025). The Evolution of Secure Communication: Analysing Cryptographic Methods from Ancient to Modern Era. International Research Journal.
[5] Globus Toolkit Documentation. Security and Encryption Concepts in Distributed Systems.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY