Journal of Population Therapeutics & Clinical Pharmacology

RESEARCH ARTICLE
DOI: 10.53555/jptcp.v30i13.3535

SECURE FILE TRANSMISSION SYSTEM

Sayonee Bhumgara[1*], Abhishek Tekavade[2], Dr. Anupama S. Budhewar[3]

[1*]MIT-ADT University MIT ADT Campus, Rajbaugh, Loni Kalbhor – 412201 +91 9021811479
Email:-bhumgarasayonee@gmail.com
[2]MIT-ADT University MIT ADT Campus, Rajbaugh, Loni Kalbhor – 412201 +91 9922839147
Email:-abhishektekavade@gmail.com
[3]MIT-ADT University MIT ADT Campus, Rajbaugh, Loni Kalbhor – 412201 +91 97649 22888
Email:-anupama.budhewar@mituniversity.edu.in

***Corresponding author: -** Sayonee Bhumgara
*MIT-ADT University MIT ADT Campus, Rajbaugh, Loni Kalbhor – 412201 +91 9021811479
Email:-bhumgarasayonee@gmail.com

Abstract
In an era where data security reigns supreme, the Secure File Transmission System emerges as an indispensable digital solution. It caters to users in search of a flawless and secure file transfer experience while addressing the ever-pressing demands for efficiency and user-friendliness. Above all, it champions the cause of safeguarding the utmost confidentiality and integrity of transmitted data.
Our research journey takes a deep dive into the inception, design, and meticulous implementation of the Secure File Transmission System. Through this exploration, we unravel its unique attributes that set it apart as a crucial tool. This system transcends the boundaries of individuals and organizations, offering a dependable shield against the challenges posed by secure data transmission in the dynamic digital landscape.
In a world where data breaches loom large, the Secure File Transmission System stands as a bastion of trust and reliability, assuring users of seamless, secure, and confidential data transfer.

Keywords:- Secure File Transmission; Data Security; Encryption Techniques; Network Security; Confidentiality and Integrity; Efficient File Transfers; Cyber Threats Trust and Reliability; Secure Data Transfer

1. INTRODUCTION

In the present digital landscape, the significance of secure data sharing cannot be overstated. Our research endeavors are dedicated to the creation of a robust Secure File Transfer System, catering to the ever-growing need for secure data transmission. To achieve this, we employ a cutting-edge technology stack featuring Node.js, Express.js, MongoDB Atlas, Google OAuth 2.0, crypto.js, Crypto, all hosted on the Heroku platform.
Our chosen approach revolves around efficient data storage on the internet, with a focus on ensuring data accessibility from anywhere, at any time. MongoDB Atlas, a fully managed database solution, is our storage platform of choice, tailored to support modern applications.
To enable real-time, bidirectional, and event-driven communication, we harness the versatility of Socket.IO. This technology seamlessly operates across diverse platforms, browsers, and devices,

serving as the backbone for instant messaging within our system.
Heroku, a prominent platform-as-a-service (PaaS) provider, empowers us to seamlessly deploy, manage, and scale our application.
Security is at the forefront of our system's design. To achieve this, we implement the AES encryption algorithm (256-bit), celebrated for its swiftness and unwavering security, setting it apart from other symmetric encryption methods.
In essence, our mission is to deliver a secure web application functioning as a Secure File Transfer System. It not only offers encryption, storage, and sharing capabilities but also guarantees the utmost data confidentiality and integrity, meeting the ever- evolving demands of secure data transmission.

2. LITERATURE REVIEW

2.1 Secure File Storage & Sharing on Cloud UsingCryptography

Authors: Madhumala RB, Sujan Chhetri, Akshatha KC,Hitesh Jain
Year of Publication: 2021
Objective:A
Authenticate users securely using GoogleOauth2.0.
Provide secure access to stored files for authenticatedusers.
Enable secure transmission of files between users. Securely share decryption passwords through a chat client.Store data in encrypted form on the cloud.
Establish secure communication between users.

Contribution:
Web application utilizing cryptography for data protection. Technology stack selection, cloud computing model, database management, real-time communication, encryption, and Heroku deployment.
Enhanced privacy and security features, including time- limited shareable links and multiple authentication options.

2.2 Efficient and Secure File Transfer in Cloud Through Double Encryption Using AES and RSA Algorithm

Authors: K. Jaspin, Shirley Selvan, Thanmai.G
Year of Publication: 2021
Objective:
Enhance data security in cloud storage services,specifically in platforms like Dropbox.
Utilize double encryption with AES and RSA algorithmsto ensure confidentiality and integrity.
Provide a higher level of protection while maintainingefficiency and speed.

Contribution:
Introduction of Double Encryption Technique using AESand RSA.
Key generation, parameter evaluation, and security analysis.
Faster runtime and improved data protection compared to existing encryption methods.

2.3 Secure and Efficient Lightweight Symmetric Encryption Scheme for Transfer of Text Files between Embedded IoT Devices

Authors: Sreeja Rajesh, Varghese Paul, Varun G. Menon,Mohammad R. Khosravi
Year of Publication: 2019

Objective:
Address security concerns in IoT networks.
Improve upon limitations of existing encryptionalgorithms like TEA.
Develop Novel Tiny Symmetric Encryption Algorithm(NTSA) for enhanced security.

Evaluate NTSA performance in an IoT network context.

Contribution:
Introduction of NTSA for enhanced data confidentiality and reduced encryption time.
Outperformance of TEA, XTEA, and XXTEA in terms of execution times for encryption and decryption.
Potential integration of NTSA for secure data transfer in various IoT applications.

2.4 Novel Selective Encryption DWT-based Algorithm for Medical Images

Authors: Med Karim Abdmouleh, Ali Khalfallah, Med Salim Bouhlel

Year of Publication: 2017
Objective:
Enhance security of medical image transmission and storage in telemedicine.
Optimize image size for faster transmission and enhanced storage capacity.
Ensure privacy through encryption to meet ethical standards.

Contribution:
Introduction of a novel cryptocompression approach for medical images.
Partial encryption of DWT matrix components for security and compatibility with JPEG2000.
Notable advantages in terms of speed, efficiency, reduced processing time, and medical image encryption resilience.

2.5 Design of a Secure File transfer System Using Hybrid Encryption Techniques

Authors: Abdeldime M.S. Abdelgader, Lenan Wu, Mohamed Y. E. Simik, Asia Abdelmutalab
Year of Publication: 2015
Objective:
Develop a security system for file transfer over the internet or communication networks.
Provide privacy, integrity, and authentication for sensitive information exchange.
Utilize modern encryption algorithms like AES, IDEA, and RSA for efficiency and speed.
Contribution:
Introduction of a two-way secure file transfer system.
Utilization of multiple encryption algorithms, hash functions (MD5), key generation (RSA), compression, and splitting for enhanced security.
Flexibility for accommodating other cryptographic systems in the future.

3. SYSTEM ARCHITECTURE

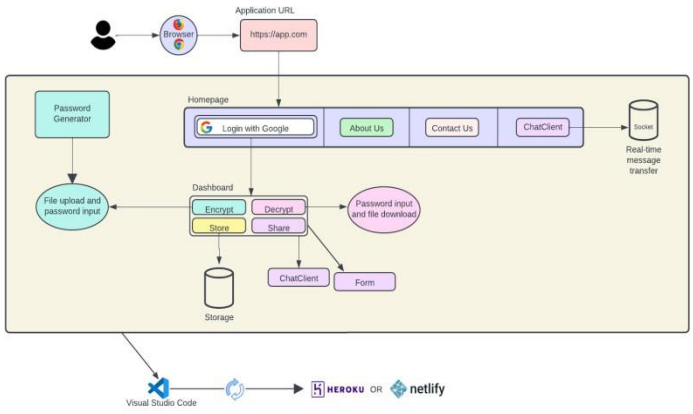


Fig. 1 Architecture diagram for the proposed system

3.1 Frontend:

3.1.1 **User Interface (UI):** The frontend provides the graphical user interface accessible through a web browser. It includes pages for user login, two-step authentication, an "About Us" page, and a "Contact Us" page.

3.1.2 **User Authentication:** Users log in through the login page, where they enter their credentials. Two-step authentication verifies the user's identity, adding an extra layer of security.

3.1.3 **Dashboard:** After successful authentication, users are directed to the dashboard on the home page. The dashboard is the central hub for file-related actions.

3.1.4 **WebSocket:** WebSocket is used to facilitate real-time messaging and notifications between the frontend and the backend. This enables instant updates on file sharing or other activities.

3.2 Backend:

3.2.1 **Web Server:** The web server serves as an intermediary between the frontend and backend, handling HTTP requests and responses. It routes user requests to the appropriate backend components.

3.2.2 **Authentication Logic:** The backend manages user authentication and two-step verification. It verifies user credentials and maintains session data.

3.2.3 **File Processing Logic:** This component handles file-related operations, including encryption, decryption, storage, and sharing. When a user encrypts a file, the backend invokes the password generator to create a strong encryption key.

3.2.4 **Password Generator:** The password generator generates strong passwords that are used as encryption keys for securing files.

3.2.5 **Database:** All user data, including files, user profiles, encryption keys, and file sharing permissions, is stored in the database. The database ensures data persistence and retrieval. You can use a database management system like PostgreSQL or a similar solution.

3.3 Integration and Flow:

1) The user logs in through the frontend, where two-step authentication verifies their identity.
2) After login, users can access the "About Us" and "Contact Us" pages for information.
3) The WebSocket connection is established for real-time communication, allowing users to receive instant notifications and updates.
4) Users interact with the dashboard to perform file operations, including encryption, decryption, storage, and sharing.
5) When a file is encrypted, the backend generates a strong password via the password generator and stores the encrypted file and associated metadata in the database.
6) File sharing updates database records to grant access to specific users and sends notifications via WebSocket for real-time awareness.
7) The VS Code environment is used for programming and developing the system.
8) Heroku is used as the hosting platform to deploy the website and its components.

This architecture provides a secure and responsive platform for secure file transmission, incorporating user authentication, real- time messaging, and encryption, all backed by a robust database. It ensures the confidentiality and integrity of user data while enabling seamless user interactions.

4. IMPLEMENTATION DETAILS

4.1 SSL/TLS for Secure File Transfer:

Technical Details –

Encryption: SSL/TLS provides strong encryption for data in transit, ensuring that files are securely transmitted over the network. It uses a combination of symmetric and asymmetric encryption for data security.

Authentication: SSL/TLS can verify the identity of both the server and the client, ensuring that

you're communicating with the intended party.
Widely Supported: SSL/TLS is supported by most programming languages and platforms, making it a versatile choice for secure file transfer across different systems.
Standard Protocol: SSL/TLS is an established and standardized protocol for securing communication over the internet, making it a trusted option.
Variety of Use Cases: SSL/TLS can be used in various ways, including securing FTP (FTPS), HTTP (HTTPS), and email (SMTP/IMAP/POP3) protocols.

4.2 SSH (Secure Shell) for Secure File Transfer Implementation:

Technical Details –
Protocol: SSH is employed as a secure network protocol dedicated to encrypted file transfer.
Extension: SFTP (SSH File Transfer Protocol) is chosen as the extension for secure file transfers.
Authentication: Key pairs (public and private keys) form the basis for authentication, with the option to include additional methods.
Encryption Algorithms: Robust encryption algorithms, such as AES, are utilized to ensure the secure transmission of data.
Access Controls: Meticulous configuration of access controls, auditing mechanisms, and firewall rules enhances overall security.
Client Applications: Secure SFTP client applications like OpenSSH and WinSCP are employed to establish secure connections.
Benefits: Using SSH/SFTP guarantees the confidential and secure transfer of files, incorporating encryption and robustauthentication.

5. LIMITATIONS OF AES ENCRYPTION

1. **Performance Variability:**
- AES encryption can exhibit performance variability depending on the specific encryption mode and key size used. Some modes may have lower performance compared to others, and the choice of key size can impact both the encryption speed and the strength of security.
2. **Limited to Encryption:**
- AES is primarily focused on encryption, providing confidentiality for data at rest or in transit. However, it does not address other aspects of data security, such as authentication, access control, or data integrity. To achieve comprehensive data security, additional measures and protocols are often required.
3. **Key Management:**
- The security of AES encryption heavily depends on the management of encryption keys. If encryption keys are lost, compromised, or not properly managed, it can lead to data security breaches. Effective key management, including key generation, storage, and rotation, is crucial for AES to be effective.
4. **Lack of Forward Secrecy:**
- Forward secrecy is the property that ensures that even if an attacker obtains a current encryption key, they cannot decrypt previously captured encrypted data. AES does not inherently provide forward secrecy. Achieving this security property often requires additional key exchange mechanisms or protocols, such as Diffie-Hellman key exchange.
5. **Resource Intensive:**
- AES encryption can be resource-intensive, especially in constrained environments such as embedded systems or IoT devices. The computational overhead of AES can impact system performance and power consumption.
6. **Brute Force Attack:**
- While AES is highly secure, it is not immune to brute force attacks. In a brute force attack, an attacker systematically tries all possible encryption keys until the correct one is found. The time required for a successful brute force attack depends on the key size; longer key sizes provide higher security but require more computational effort to break.

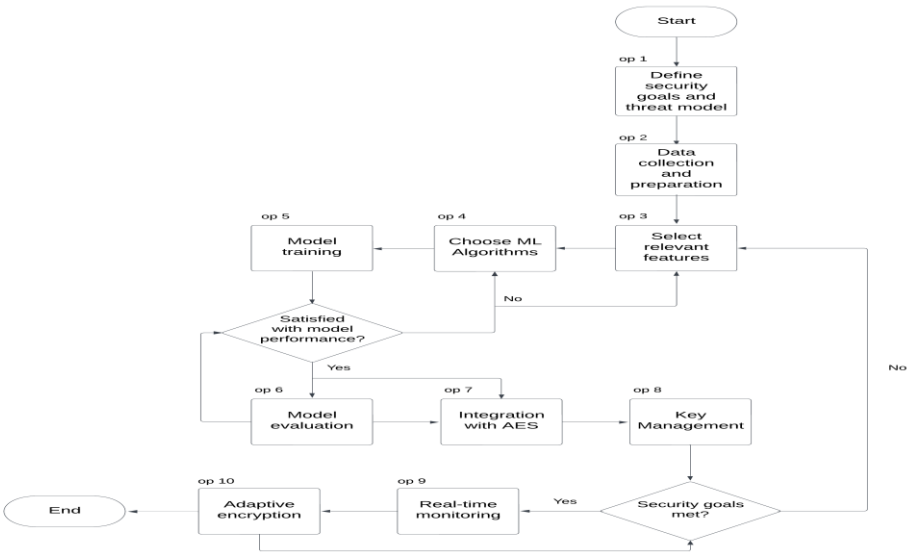
COMBINING AI WITH AES FOR IMPROVED DATA SECURITY

By integrating AI, specifically machine learning algorithms like k-Nearest Neighbors (k-NN), with AES encryption, data security is enhanced in several ways. AI can adapt to emerging threats and vulnerabilities, provide real-time threat detection, and offer behavioral analysis for anomaly detection. This integration enables a dynamic and adaptive data security solution that strengthens AES's static encryption with AI-driven intelligence.

The AI component can identify anomalies in encrypted data, recognize patterns of malicious activity, and respond to security threats as they occur, addressing some of the limitations of AES. The result is a comprehensive and adaptive data security system that combines the encryption strength of AES with the real-time threat detection and adaptive response of AI.

AI, with its machine learning algorithms, brings adaptability, pattern recognition, and real-time threat detection to the table. In particular, k-NN excels in recognizing patterns and anomalies in data, making it a valuable addition to AES encryption.

Our research aims to demonstrate how the dynamic adaptability of k-NN can complement AES, improving real-time threat mitigation and providing a safeguard against emerging vulnerabilities. The k-NN algorithm can detect anomalies in encrypted data, offering a vital defense against threats that may otherwise go unnoticed.



Flowchart: Integration of Machine Learning algorithms with Advanced Encryption Standard

6. RESULTS

6.1.1 Time Improvement in Encryption –

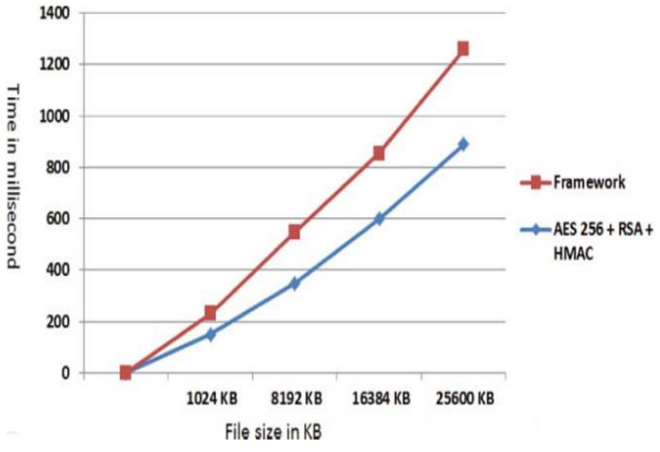


Diagram 6.1.1 *Performance evaluation of encryption process using proposed framework*

The above figure shows the performance evaluation of the proposed framework against AES 256 combined with RSA and HMAC.

Size	AES 256 + RSA + HMAC	Framework	Improvement
1024 KB	150	80	46.666% decreased
8192 KB	350	198	43.428% decreased
16,384 KB	600	255	57.499% decreased
25,600 KB	890	369	58.539% decreased
51,200 KB	1220	480	60.655% decreased
Average	642	276.4	56.947% decreased
Throughput KB/ms	31.900 KB/ms	55.756 KB/ms	74.783% increased

Table 6.1.1 *Time required for encryption process (milliseconds)*

6.1.2 Time Improvement in Decryption –

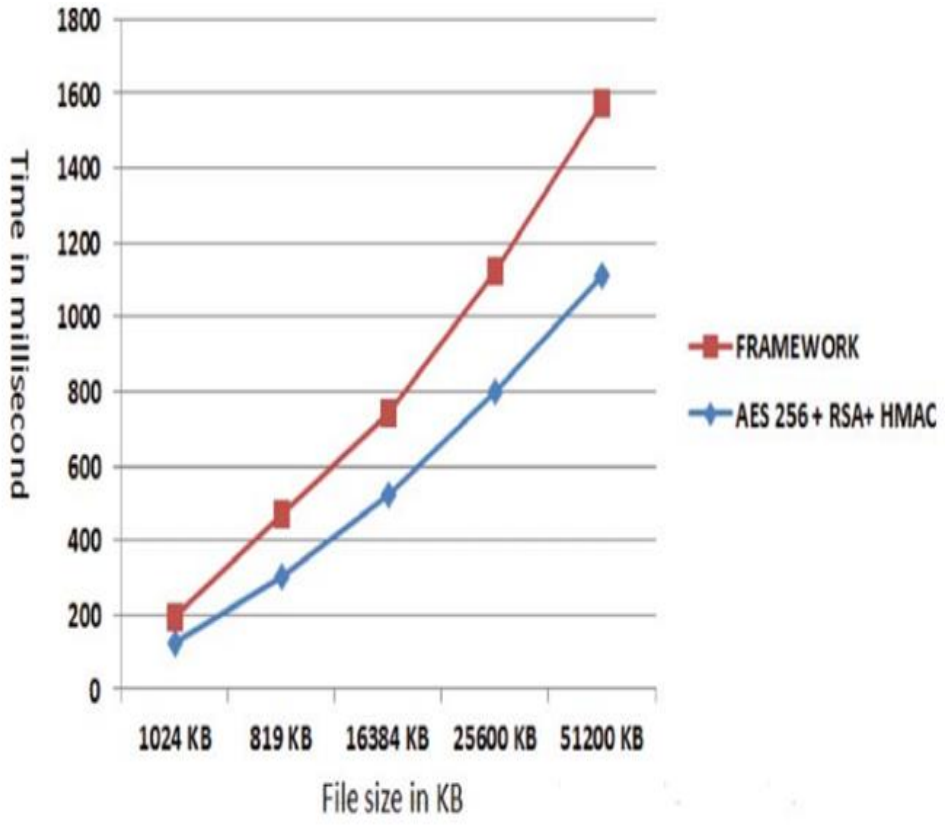


Diagram 6.1.2 *Performance evaluation of decryption process using proposed framework*

Size	AES 256 + RSA + HMAC	Framework	Improvement
1024 KB	122	71	41.803% decreased
8192 KB	305	168	44.918% decreased
16,384 KB	520	221	57.5% decreased
25,600 KB	798	325	59.273% decreased
51,200 KB	1110	465	58.108% decreased
Average	571	250	56.217% decreased
Throughput KB/ms	26.989 KB/ms	61.644 KB/ms	128.404% increased

Table 6.1.2 *Time required for decryption process (milliseconds)*

6.1.3 Memory Improvement in Encryption -

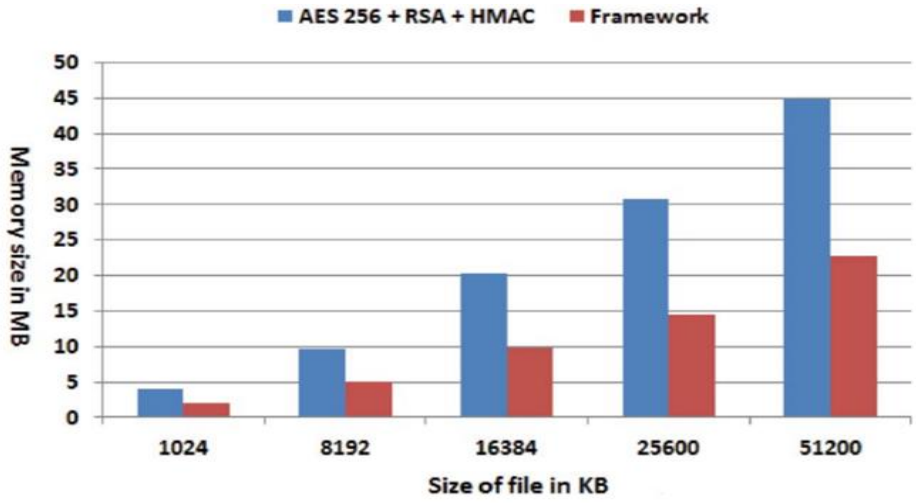


Diagram 6.1.3 *Memory usage in encryption process*

Size	AES 256 + RSA + HMAC	Framework	Improvement
1024	3.92	2.01	48.754 decreased
8192	9.66	5.1	47.204 decreased
16384	20.2	9.91	50.940 decreased
25600	30.72	14.48	52.864 decreased
51200	44.81	22.68	49.386 decreased

Table 6.1.3 *The memory utilization in encryption process*

6.1.4 Memory Improvement in Decryption –

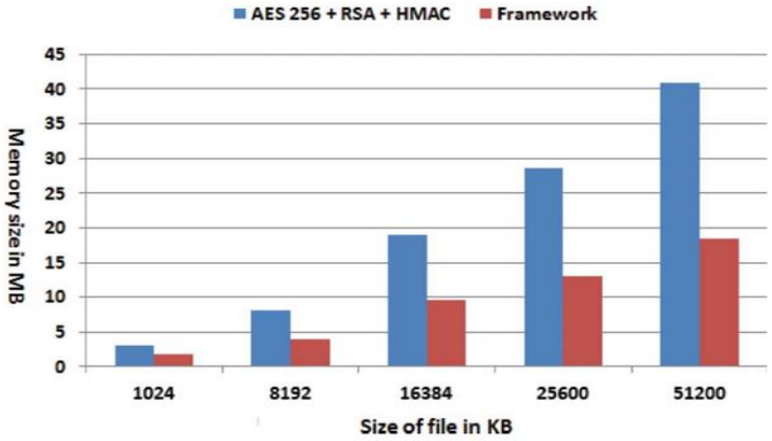


Diagram 6.1.4 *Memory usage in decryption process*

Size	AES 256 + RSA + HMAC	Framework	Improvement
1024	3.1	1.71	44.838 decreased
8192	8.05	3.98	50.559 decreased
16,384	18.92	9.64	49.048 decreased
25,600	28.54	13.09	54.134 decreased
51,200	40.88	18.49	54.770 decreased

Table 6.1.4 *the memory utilization in decryption process*

7. CONCLUSION

In conclusion, the Secure File Transmission System (SFTS) project aligns with market demand for secure file transfer solutions. Competitor analysis, market acceptance, and adaptation to emerging trends underline its strategic positioning. Income generation ideas ensure sustainability, and a commitment to affordability aims to make SFTS accessible to a broad user base. This project stands poised to provide a reliable and secure solution for users seeking to protect their data during transmission

8. ACKNOWLEDGMENT

We want to extend our vote of thanks to our project guide **Prof. Dr. Anupama Bhalerao** for helping us and guiding us wherever possible. We would also like to thank our college MIT ADT University for providing us the appropriate resources and the format of the research paper. Without our guide and the collegewe would not have been able to produce a work of this quality.

9. REFERENCES

1. Madhumala RB, Sujan Chhetri, Akshatha KC, Hitesh Jain, 2021, “Secure File Storage & Sharing on Cloud Using Cryptography”.
2. K. Jaspin, Shirley Selvan, Thanmai.G, 2021, “Efficient and Secure File Transfer in Cloud Through Double Encryption Using AES and RSA Algorithm”.
3. Sreeja Rajesh, Varghese Paul, Varun G. Menon, Mohammad R. Khosravi, 2019, “Secure and Efficient Lightweight Symmetric Encryption Scheme for Transfer of Text Files between

Embedded IoT Devices".
4. [4] Med Karim Abdmouleh, Ali Khalfallah, Med Salim Bouhlel, 2017, "Novel Selective Encryption DWT-based Algorithm for Medical Images".
5. Abdeldime M.S. Abdelgader, Lenan Wu, Mohamed Y. E. Simik, Asia Abdelmutalab, 2015, "Design of a Secure File transfer System Using Hybrid Encryption Techniques".
6. U. A. Solomon, Raj C. P. Maheswaran, 2023, "Secure File Sharing System Using Image Steganography and Cryptography Techniques."
7. Wanzong Peng, Tongliang Lu, Zhongpan Wang, 2023, "Blockchain-Based File Transfer Framework."
8. K. Jaspin, Shirley Selvan, Sahana Sridhar, Thanmai Gutta, 2021, "Efficient and Secure File Transfer in Cloud Through Double Encryption Using AES and RSA Algorithm."
9. Sam Banani Surapa, Thiemjarus KittiWongthavara, Nattapong Ounanong, December 2021, "A Dynamic Lightweight Symmetric Encryption Algorithm for Secure Data Transmission via BLE Beacons (Bluetooth Low Energy)."
10. Madhumala RB, Sujan Chhetri, Akshatha KC, Hitesh Jain, 2021, "Secure File Storage & Sharing on Cloud Using Cryptography.
11. Felipe Cunha, et al., June 2016, "Data Communication in VANETs: Protocols, Applications, and Challenges.
12. Abdeldime M.S. Abdelgader, et al., October 2015, "Design of a Secure File Transfer System Using Hybrid Encryption Techniques.
13. Anupriya Aggarwal and Praveen Kanth, January 2014, "Secure Data Transmission Using DNA Encryption.
14. Rajan S. Jamgekar and Geeta Shantanu Joshi, February 2013, "File Encryption and Decryption Using Secure RSA.
15. Xubin Li, et al., 2012, "Performance Evaluation of Symmetric Encryption Algorithms.
16. Rachna Arora and Anshu Parashar, August 2013, "Secure User Data in Cloud Computing Using Encryption Algorithms
17. Murali Krishna, et al., April 30th, 2011, "Secure File Multi Transfer Protocol Design."
18. Diaa Salama Abdul Elminaam, et al., May 2010, "Performance Evaluation of Symmetric Encryption Algorithms.
19. Tiegang Gao and Zengqiang Chen, 2007, "A New Image Encryption Algorithm Based on Hyper-Chaos.
20. Lawrie Brown and Martin Gilje Jaatun, December 1992, "Secure File Transfer Over TCP/IP."