**Secure File Transmission System**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  | |  |
| Sayonee Bhumgara  MIT-ADT University  MIT ADT Campus, Rajbaugh  Loni Kalbhor - 412201  +91 9021811479  bhumgarasayonee@gmail.com    Jayditya Nandkar  MIT-ADT University  MIT ADT Campus, Rajbaugh  Loni Kalbhor - 412201  +91 9285533055 jayditya2082@gmail.com | Abhishek Tekavade  MIT-ADT University  MIT ADT Campus, Rajbaugh  Loni Kalbhor - 412201  +91 9922839147  abhishektekavade@gmail.com    Dr. Anupama S. Budhewar  MIT-ADT University  MIT ADT Campus, Rajbaugh  Loni Kalbhor - 412201  +91 97649 22888  anupama.budhewar@mituniversity.ed | | Atharva Zanwar  MIT-ADT University  MIT ADT Campus, Rajbaugh  Loni Kalbhor - 412201  +91 7887448502 athzanwar@gmail.com |
|  | u.in |
| **ABSTRACT** |  | 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 everevolving demands of secure data transmission.     1. **LITERATURE REVIEW**   **2.1 Secure File Storage & Sharing on Cloud Using Cryptography**    **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 authenticated users. | |
| 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. | |

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 timelimited 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 algorithms to ensure confidentiality and integrity.

Provide a higher level of protection while maintaining efficiency and speed.

**Contribution:**

Introduction of Double Encryption Technique using AES and 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 encryption algorithms 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.

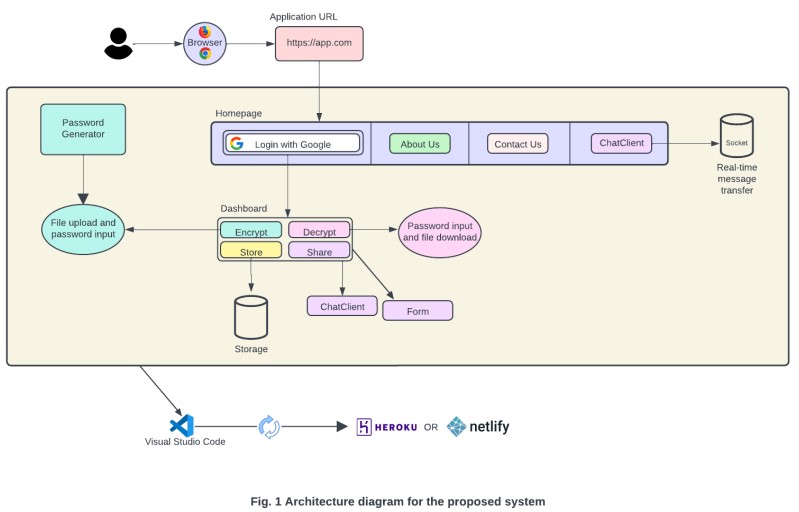
**Contribution:**

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.

# SYSTEM ARCHITECTURE



**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.

**3.1.2User 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 realtime 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, realtime messaging, and encryption, all backed by a robust database. It ensures the confidentiality and integrity of user data while enabling seamless user interactions.

# 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 robust authentication.

# 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

# 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 college we would not have been able to produce a work of this quality.

# 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. 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”.