Secure File Sharing System Using Proxy Re-Encryption

Sameer Khan
Dept. of computer science
Bowling Green State University (BGSU)
Bowling Green, OHIO, USA, 43403
Sameerk@bgsu.edu

***Abstract*—In today's digital age, the secure and efficient sharing of sensitive information is of utmost importance. This project aims to develop a Secure File Sharing System that utilizes advanced proxy re-encryption methods to ensure robust data security and flexible access control. The system employs a sophisticated cryptographic framework to facilitate secure encryption, transmission, and decryption of files, effectively mitigating risks of unauthorized access. The architecture of the system is designed around secure user authentication mechanisms, role-based access controls, and a Java-based backend that handles complex encryption and decryption processes. The system is complemented by a dynamic web interface that uses JavaServer Pages (JSP), allowing for user-friendly functionalities such as file uploads, downloads, and secure access management.**

By using proxy re-encryption, the system allows files to be securely re-encrypted and accessed by new users, while keeping the underlying data hidden. This approach enhances both security and usability, making it more difficult for cyber threats to compromise the system. The implementation showcases a practical solution for secure file sharing within organizations, highlighting its potential to revolutionize data protection strategies in various business and technology landscapes.

1. Introduction

In today's digital world, sharing data across different platforms has become increasingly common. This has made it crucial to have strong security measures in place, especially when it comes to file sharing. Although traditional encryption methods have been effective to some extent, they often fall short in terms of managing access controls in a flexible and efficient manner. This challenge is made even more complex by the need for systems that can adjust to the various roles and permissions of different users, while still ensuring the confidentiality of the shared data.

This project aims to create a Secure File Sharing System that can ensure the confidentiality and integrity of data both in transit and at rest. The system is designed to provide flexible and dynamic access control while utilizing the proxy re-encryption method. Authorized users will be able to access and re-encrypt files securely without revealing the underlying plaintext data or requiring multiple rounds of decryption and re-encryption. This approach significantly reduces the overheads associated with traditional encryption methods and enhances the overall security framework.

The project aims to create a system architecture that provides a user-friendly web interface, backend encryption services, and access management, along with database support for user and file data. The implementation will use Java and JavaServer Pages (JSP) to ensure a seamless and secure user experience, starting from registration and authentication, all the way to file management and access control.

2. Related Work

In the rapidly evolving field of digital security, proxy re-encryption (PRE) stands out as a key technique for enhancing secure data sharing across cloud and IoT platforms. This section examines key contributions to PRE, showcasing how researchers have continually refined this technology to meet modern security demands effectively.

2.1. Blockchain-Based Secure Data Sharing

Liu et al. (2023) propose a novel method integrating proxy re-encryption (PRE) and blockchain to enhance secure data sharing. Their approach addresses cloud storage limitations and single-point failure issues by storing encrypted data on IPFS and embedding digital watermarks for data leak traceability. Leveraging attribute-based encryption (ABE), they track users sharing decryption keys, ensuring secure sharing and data traceability. Employing threshold-based PRE, distributed among multiple proxy nodes, enhances security and mitigates collusion attacks. Their scheme is validated through a security analysis, demonstrating resilience against collusion and passive attacks, meeting cryptographic security standards. In summary, their blockchain-based solution sets a new standard for secure, accountable data sharing [1].

2.2. User-End Encrypted Cloud Data Sharing

Hu, Wong, and Tang (2016) delve into enhancing secure data sharing in cloud environments. They introduce innovative proxy re-encryption (PRE) schemes devoid of pairings, significantly boosting computational efficiency and public verifiability. Their CCA-secure, single-hop unidirectional PRE scheme enables a semi-trusted proxy to convert ciphertext between users without revealing plaintext or necessitating multiple encryptions. Their approach marks a significant departure from traditional bilinear pairing-based methods, offering enhanced efficiency and public verifiability. Through rigorous analysis, they demonstrate the schemes' security under chosen-ciphertext attacks (CCA) and their superior performance compared to existing literature. They guided further research in constructing CCA-secure PRE schemes without pairings, underscoring the ongoing need for advancements in this domain [2].

2.3. Efficient Key-Aggregate Proxy Re-Encryption

Chen, Fan, and Tseng (2018) introduce a groundbreaking solution for secure data sharing in cloud environments: the key-aggregate proxy re-encryption (KAPRE) scheme. By merging key-aggregate cryptosystems with proxy re-encryption, they create a framework where re-encryption keys remain constant in size regardless of data sharing conditions. This approach enables secure delegation of access to encrypted files without user online presence, offloading computational burden to the cloud. Their system addresses key issues in previous schemes, such as abuse of re-encryption keys and fine-grained access control, ensuring data security and preventing unauthorized access. Through theoretical and empirical analyses, they confirm the security and efficiency of their scheme, setting a new standard for scalable and secure data sharing in cloud computing [3].

2.4. CCA Secure Proxy Re-Encryption Scheme

In their paper, Mishra and Jena (2018) tackle the security challenges posed by quantum computing to proxy re-encryption schemes. They introduce a novel CCA-secure proxy re-encryption scheme resistant to quantum attacks, leveraging the Ring-LWE problem. Their system enhances cloud-based file sharing by ensuring both re-encryption security and data confidentiality with efficient processing times. It provides anonymity in re-encryption transactions, safeguarding parties' identities. Through rigorous security analysis, the scheme demonstrates superior computational efficiency and robustness under chosen ciphertext attacks. Mishra and Jena's work sets a new standard for secure data sharing, offering advanced solutions that prioritize privacy and efficiency in cloud storage services [4].

2.5. Fine-Grained Data Sharing in IIoT

Zhang et al. (2024) address secure and efficient data-sharing challenges in the Industrial Internet of Things (IIoT). They propose a fine-grained data-sharing scheme using proxy re-encryption (PRE) to lessen the computational burden on data owners while ensuring robust security. Leveraging identity-based encryption (IBE) and attribute-based encryption (ABE), their scheme is well-suited for resource-constrained environments. By offloading heavy computations to a semi-trusted proxy server, they maintain security while enhancing efficiency. Formal security proofs attest to its resistance to chosen-plaintext attacks. Performance evaluations demonstrate its superiority in reducing computational and communication overhead compared to traditional methods. Zhang et al.'s approach offer a practical, scalable solution for real-world IIoT applications, catering to dynamic industrial data-sharing needs [5].

My project is upon the foundation laid by previous research on proxy re-encryption, which has helped advance secure data sharing across digital platforms. By integrating user-centric design with advanced cryptographic techniques, I refine proxy re-encryption to create a Secure File Sharing System. This system not only addresses data privacy and access control challenges but also reduces the overhead of traditional encryption methods. By combining established knowledge with innovations, my work contributes to the ongoing conversation on secure digital communications and introduces practical solutions with potential for future impact.

3. Methodology

This section delves into the methodology used to implement the Secure File Sharing System.

3.1. System Design and Architecture

The Secure File Sharing System was developed through a comprehensive design phase, where the system architecture was carefully planned to support scalability, security, and efficient data management. The architecture consists of three primary components: the web interface, the backend server, and the database. The web interface provides a user-friendly platform for easy file management tasks like uploading, downloading, and access control. The backend server handles encryption, decryption, and user authentication. The database securely stores user credentials, access permissions, and file data.

3.2. Cryptographic Algorithms

The Secure File Sharing System uses a cryptographic framework that guarantees the confidentiality, integrity, and availability of data. This framework makes use of sophisticated cryptographic algorithms and techniques, including proxy re-encryption. Proxy re-encryption is a critical component that enables secure and efficient access management across distributed systems.

3.2.1. Proxy Re-encryption Scheme. Proxy re-encryption is a type of public-key encryption that permits a proxy entity to convert ciphertexts from one public key to another, without gaining any knowledge of the underlying plaintext. This ability is crucial to the system's capability of enabling file owners to share encrypted data with other authorized users in a secure manner, without the need to manually decrypt and re-encrypt the data. The usage of proxy re-encryption reduces the risks associated with key distribution and minimizes exposure to sensitive data.

- **Key Generation:** unique public/private key pairs are generated for each user. The proxy uses these keys to perform re-encryptions.
- **Re-encryption Process:** When a file owner wishes to grant access to another user, the proxy converts the ciphertext encrypted with the owner's public key into a form that can be decrypted by the recipient's private key.
- **Security Benefits:** This method prevents the proxy from accessing the unencrypted data, ensuring that the re-encryption process is secure and transparent.

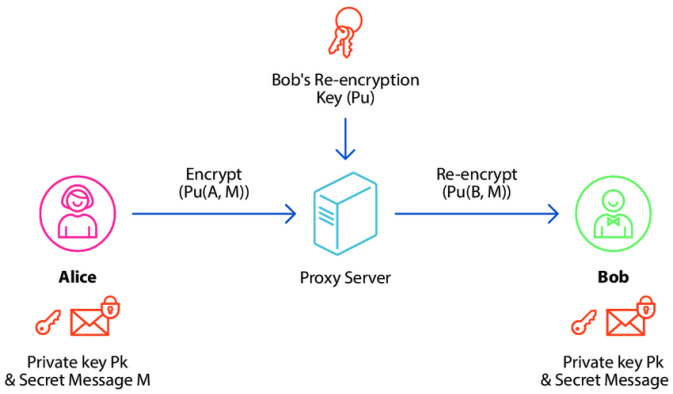

Figure 1. Overview of proxy re-encryption scheme. [5]

Example (shown in Fig.1)

The system architecture incorporates three principal entities:

- **Data Owner (Alice):** Initially, Alice encrypts her file using her private key (Pk) and a secret message (M). The encrypted file is denoted as Enc(Pu(A),M), where Pu(A) is Alice's public key, and M is the message or file content.
- **Proxy Server:** Once the file is encrypted, it is uploaded to a cloud-based proxy server. The proxy server is the pivotal component that facilitates the PRE process without ever decrypting the file.
- **Recipient (Bob):** To share the encrypted file with Bob, Alice provides the proxy server with Bob's re-encryption key (Pu). The proxy server uses this key to re-encrypt the file for Bob, transforming it into Re-Enc(Pu(B),M), which Bob can decrypt using his private key.

3.2.2. Encryption and Decryption Processes. The encryption and decryption processes are designed to be both efficient and secure, meeting the needs of a dynamic and distributed user base.

- **Encryption:** Files are encrypted using a symmetric key algorithm for speed and efficiency. The symmetric key is generated securely when the file is uploaded and is then encrypted with the user's public key, ensuring that only the user can access it.
- **Decryption:** To decrypt a file, a user's private key is used to decrypt the symmetric key, which in turn is used to decrypt the file. This two-tier encryption mechanism enhances security by combining the benefits of both symmetric and asymmetric encryption.
- **Key Management:** Proper management of keys is critical. The system implements secure key storage and management policies to protect keys from unauthorized access and misuse.

3.3. Development Process

3.3.1. Planning and Design. The project began with an extensive planning and design stage. During this stage, I analyzed the requirements in detail to ensure that the developed system would meet the desired security specifications and user needs. This phase involved gathering and analyzing the requirements to align the project's objectives with the anticipated security and operational challenges. I designed a robust and modular architecture that supports scalability and integrates cryptographic processes effectively. This ensured that the system could handle various user scenarios securely.

To manage my project effectively, I used an Agile methodology. I broke down the project into smaller parts, called iterations, and focused on specific features one at a time. After each iteration, I reviewed my work and made changes to the project plan as needed.

To ensure effective development and the achievement of the system's security and functional goals, it was crucial to choose the right tools and technologies. For backend development, I selected Java due to its robust security features, extensive API support, and strong community resources. To enable dynamic content generation and smooth integration with Java's backend functionalities, I used JavaServer Pages (JSP) for the front end. For secure management of user data and file data, MySQL was the ideal choice due to its reliability and support for complex data management needs.

I tested the system thoroughly to ensure it worked well. I checked each part to fix any problems and then tested them together to confirm they worked seamlessly. I conducted tests in a simulated production environment to check the system's performance. I also performed security testing to ensure that the system was secure.

3.4. Code Flow

The Secure File Sharing System has different workflows for Data Owners, Data Users, and the Proxy Server. These workflows help maintain security and make sure the system works properly.

3.4.1. Data Owner Workflow (See Figure 2). For Data Owners, the system starts with an account request, establishing the credentials and permissions within the system. Upon successful account creation, Data Owners log in to access the system's functionalities. A key feature here is the file upload process, where the Data Owner uploads files, which are then encrypted and securely stored on the server. Subsequently, the Data Owner monitors and manages user requests for file access. When a request is considered valid, the Data Owner can initiate a re-encryption request to the Proxy Server, which is an essential step in enabling user-specific file access while maintaining data confidentiality. This request is a crucial component in the proxy re-encryption scheme, as it ensures files can be securely re-encrypted for authorized users without decrypting the files to plaintext at any intermediary stage. The Data Owner can continue to use the system's functionality or opt to log out, thereby securely terminating the session.

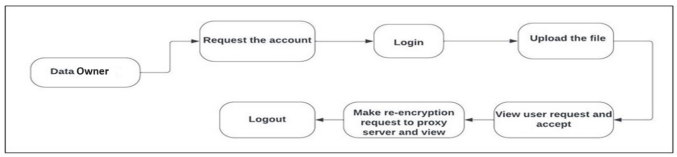

Figure 2. Data Owner Workflow

3.4.2. Data User Workflow (See Figure 3). Data Users engage with the system starting with the login procedure, which grants access to the platform based on verified credentials. The subsequent step involves searching for files necessary for the user's operations. Upon locating the desired files, users make access requests to the Data Owners. This process is crucial to the user experience, providing a transparent system for request submission and tracking. A system allows users to view the status of their access requests, which offers real-time updates and maintains user engagement. The workflow for the Data User ends with a logout.

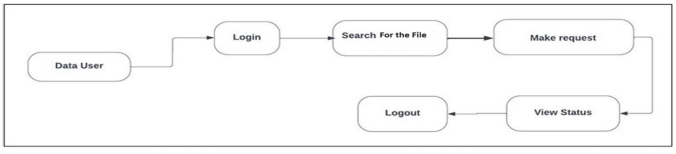

Figure 3. Data User Workflow

3.4.3. Proxy Server Workflow (See Figure 4). The Proxy Server plays the main role in the re-encryption process. The Proxy Server's workflow is straightforward yet critical, starting with a secure login to manage re-encryption requests from Data Owners. The main function of the Proxy Server is to view and process these re-encryption requests without accessing the file's content, maintaining the confidentiality and integrity of the data. The Proxy Server performs re-encryptions that allow Data Users, who have been granted access by Data Owners, to decrypt files without the need to exchange any private decryption keys. This step is essential for achieving a secure, scalable file-sharing system. The Proxy Server also ensures that operations within the system are conducted as authorized, maintaining a secure and trustworthy environment before safely logging out to complete the activities.

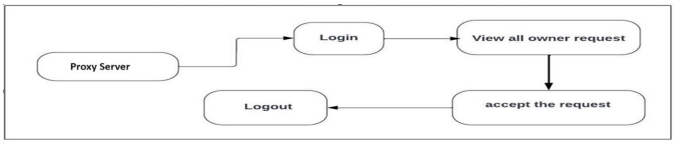

Figure 4. Proxy Server Workflow

3.5. Testing and Security Assessment

In the final stages of the development process, an extensive testing and security assessment phase was crucial to ensuring that the Secure File Sharing System met all operational and security requirements. I personally conducted thorough manual testing across various components of the system to validate functionality and identify any potential issues.

The initial phase of testing focused on verifying that the encryption mechanism was functioning correctly. This was done by examining the output files to ensure they were properly encrypted before storage, thus safeguarding the data from unauthorized access. I checked both the integrity and confidentiality aspects of the encrypted files, confirming that no readable data could be recognized without appropriate decryption.

Further, I tested the system's overall functionality by performing routine operations such as file uploading and downloading. These tests were crucial to see that the system handled file transfers smoothly and maintained file integrity throughout the process. Each file uploaded was followed by a download test to ensure that the original content was accurately retrieved, thus confirming the end-to-end security of the data transmission process.

Throughout this testing phase, documentation of all findings was maintained, which allowed for a systematic approach to addressing any issues uncovered.

4. Results

The creation and implementation of the Secure File Sharing System has resulted in a fully functional application that effectively uses proxy re-encryption to enhance file security and user accessibility. The system's capabilities and user interfaces have been documented through detailed screenshots, which illustrate its practical application and robust design.

4.1. Home and Login Interface

Figure 5 displays the system's homepage, acting as the primary entrance to the platform. It showcases separate login and registration sections for users, owners, and proxy server administrators. The interface is intentionally designed to be inviting and straightforward, ensuring seamless navigation for every user type. Data owners and users can first register and then login, whereas proxy servers do not require registration.

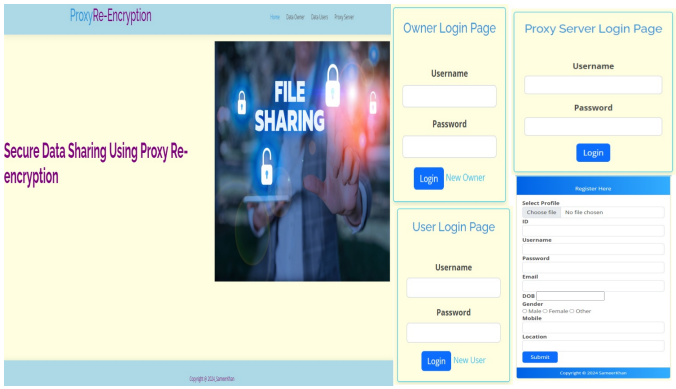

Figure 5. Home and Login Interface

4.2. Owner Home Interface

In Figure 6, we see the owner's dashboard - a central hub for managing files and ensuring their security. Using this interface, owners can easily upload files and rely on the system to automatically encrypt them using a hash key and secret key. Fields for entering file-related information also allow for comprehensive tracking and management of metadata. Notably, this dashboard also displays pending re-encryption requests from users. Owners can easily approve these requests and send them to the proxy server for processing. This integration of encryption and user management highlights the system's ability to handle sensitive information securely, without exposing actual data to the server or unauthorized parties. Finally, the data owner can log out when finished.

4.3. User Home Interface

In Figure 7, we see the user's dashboard that enables file management. Here, users can effortlessly access their

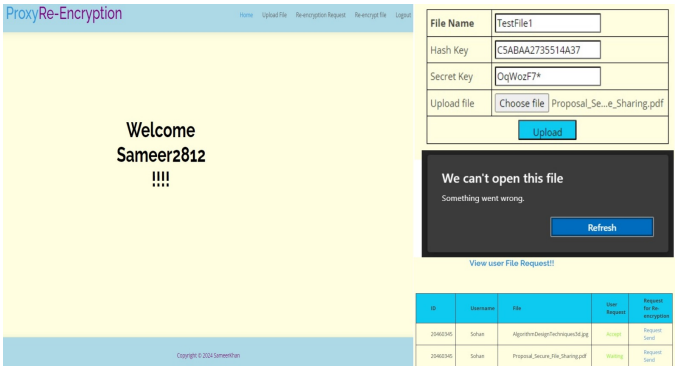

Figure 6. Owner Home Interface

profile information, request file access from owners, and securely download authorized files. This screen is pivotal in showcasing the system's end-user functionality, emphasizing its secure and regulated file access. Additionally, the interface features real-time updates on user requests, enhancing their engagement. All user actions are protected by the underlying security protocols, guaranteeing the integrity and confidentiality of the data throughout their interaction with the system.

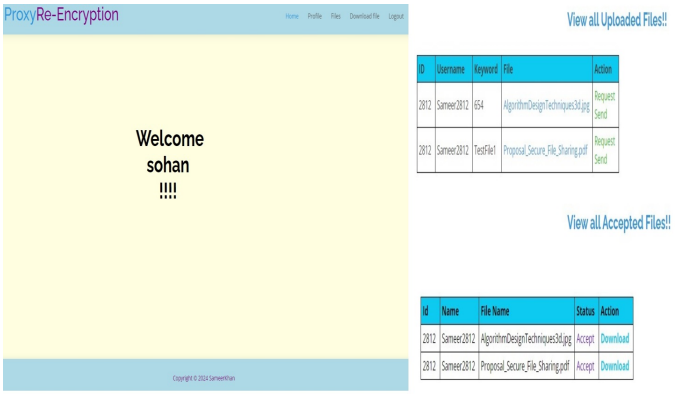

Figure 7. User Home Interface

4.4. Proxy Server Interface

In Figure 8, we see the administrative dashboard of the proxy server, which is responsible for managing essential functions such as file viewing, re-encryption processing, and key distribution to users. This interface highlights the crucial role that the proxy server plays in maintaining the system's security architecture. By utilizing proxy re-encryption in a real-world application, the proxy server can perform sensitive operations without ever accessing the actual content of the files. Additionally, the proxy server has the power to manage and authorize permissions for both users and owners, further reinforcing the system's comprehensive security measures.

The results presented show the success of the Secure File Sharing System in merging advanced cryptographic tech-

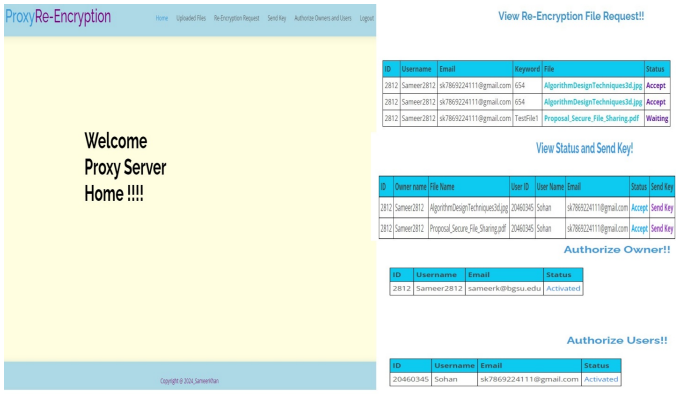

Figure 8. Proxy Server Interface

niques with a user-centric design. Through detailed screenshots and analyses, we see the system's robust functionality, secure data management, and intuitive interface, all crucial for enhancing user trust and promoting adoption.

5. Discussion

The Secure File Sharing System effectively addresses significant challenges in secure data exchange and access management by integrating proxy re-encryption. This approach enhances data security and privacy by allowing flexible file access without exposing sensitive information or the complexities of direct key exchanges. The use of proxy re-encryption not only mitigates risks associated with data breaches but also simplifies compliance with data protection standards.

Despite its benefits, the project faced challenges, including the complexity of implementing the proxy re-encryption algorithm and maintaining system performance with increasing user numbers. Additionally, the system's dependency on current cryptographic methods requires ongoing updates to address new security threats, emphasizing the need for continuous improvement in cryptographic resilience.

In summary, while the Secure File Sharing System introduces a promising solution for secure digital communications, it also highlights the need for it in cybersecurity practices to sustain its effectiveness and reliability.

6. Conclusion and Future Work

The Secure File Sharing System represents a significant advancement in the field of digital communications, offering robust security features through the implementation of proxy re-encryption. This project has successfully demonstrated how cryptographic enhancements can improve security protocols while maintaining user accessibility and system efficiency. By ensuring that only authorized users can access and re-encrypt files without compromising security, the system addresses critical gaps in traditional file-sharing models, which often struggle with secure access management and data privacy.

Looking forward, there are several paths for future work and research. Firstly, exploring the integration of more advanced and contemporary encryption algorithms, such as AES with a longer key size, could provide stronger security measures to keep pace with evolving cyber threats. Additionally, enhancing the system's architecture to support scalable cloud-based environments could extend its applicability and efficiency, particularly for enterprise-level deployments. Another promising area of development is the incorporation of machine learning techniques to predict and respond to security threats proactively, potentially automating certain aspects of security and risk management within the system.

In conclusion, while the Secure File Sharing System has laid a solid foundation for secure data exchange, ongoing research and development are crucial to adapt to the rapidly changing landscape of cybersecurity and to meet the growing demands of users for safer and more efficient digital communication solutions.

References

[1] J. Liu, C. Li, R. Wang, J. Li and W. Xia, "A Blockchain-Based Secure Data Sharing Approach with Proxy Re-Encryption," 2023 8th International Conference on Data Science in Cyberspace (DSC), Hefei, China, 2023, pp. 128-134, doi: 10.1109/DSC59305.2023.00028.

[2] X. Hu, C. Tang and D. S. Wong, "Highly Efficient Proxy Re-encryption Schemes for User-End Encrypted Cloud Data Sharing," 2016 15th International Symposium on Parallel and Distributed Computing (ISPDC), Fuzhou, China, 2016, pp. 261-268, doi: 10.1109/ISPDC.2016.45.

[3] W. -H. Chen, C. -I. Fan and Y. -F. Tseng, "Efficient Key-Aggregate Proxy Re-Encryption for Secure Data Sharing in Clouds," 2018 IEEE Conference on Dependable and Secure Computing (DSC), Kaohsiung, Taiwan, 2018, pp. 1-4, doi: 10.1109/DESEC.2018.8625149.

[4] B. Mishra and D. Jena, "CCA Secure Proxy Re-Encryption Scheme for Secure Sharing of Files through Cloud Storage," 2018 Fifth International Conference on Emerging Applications of Information Technology (EAIT), Kolkata, India, 2018, pp. 1-6, doi: 10.1109/EAIT.2018.8470404.

[5] Q. Zhang, Y. Fu, J. Cui, D. He and H. Zhong, "Efficient Fine-Grained Data Sharing based on Proxy Re-encryption in IIoT," in IEEE Transactions on Dependable and Secure Computing, doi: 10.1109/TDSC.2024.3386690.

Appendix A.
Source Code of Project

To access the Source Code Click Here Secure File Sharing System Using Proxy Re-Encryption