

Secure File Sharing System with Strong Password and One Time Password Authentication

Chee Lee Chong¹, Nur Ziadah Harun^{2*}

^{1,2}Faculty of Computer Science and Information Technology,
University Tun Hussein Onn Malaysia, 86400 Parit Raja, Johor, Malaysia

ARTICLE INFO

Article history:

Received 24 October 2024
Revised 31 December 2024
Accepted 10 January 2025
Online first
Published 1 March 2025

Keywords:

File Sharing System
Cloud Storage
Multi-Factor Authentication
One-time password

DOI:

10.24191/jcrinn.v10i1.500

ABSTRACT

A file-sharing system is a system that allows users to share their files with other users. In the digital age, file-sharing systems have become integral for various sectors, including corporate, educational, and creative fields. Inadequate security measures in file-sharing systems have become a significant concern. Many current systems expose users to unauthorized access and data breaches due to weak authentication methods, such as using simple username and password combinations. Additionally, the lack of a secure password when files are shared through links exacerbates the risk, as unauthorized users can access confidential data without needing a password. This paper presents the development of a secure file-sharing system incorporating strong password protocols and one-time password (OTP) authentication to address vulnerabilities in existing systems. The proposed system ensures robust security by implementing multi-factor authentication, end-to-end encryption, hashing in database and secure session management. The users must provide a strong password and OTP for login, register and reset password process. Key features include user management and comprehensive file management functionalities. The system offers user account management, space management, and user file management, ensuring that administrative controls are in place to manage user activities and storage efficiently. The developed system implements the file management module such as upload, download, rename, move, create folders, and share with password protection. The implementation details and testing results confirm the system's effectiveness in providing a secure and user-friendly file-sharing platform. At the end of this project, the proposed system has achieved the objective of developing a file-sharing system that can share files securely.

1. INTRODUCTION

File sharing is a process offering access to digital information or resources such as documents, images, audio, and e-books to other people. This process permits others to use digital stuff either privately among a group or publicly to everyone. This system enables the users to upload, access, and share files over the internet. These systems play a vital role in diverse sectors. In corporate settings, these platforms optimize

^{2*} Corresponding author. *E-mail address:* nurziadah@uthm.edu.my
<https://dx.doi.org/10.24191/jcrinn.v10i1.500>

workflow, allowing employees to collaborate efficiently and share essential documents swiftly (Kulkarni et al., 2012). Educational institutions rely on these systems to create seamless learning experiences, fostering interactive environments for students. Furthermore, in creative fields, file-sharing platforms enable the exchange of multimedia content, empowering artists, musicians, and designers to collaborate on a variety of projects effectively.

The objective of this project is to propose and develop a secure file sharing system with file encryption using strong password and one-time password authentication, to develop a secure file sharing system with file encryption using strong password and one-time password authentication. The scope of the project is developing a file-sharing system that consists of the signup, login, upload, download, transfer, and encryption of file modules.

2. INTRODUCTION

In this section, the study domain, employed technology, and outcomes of the comparative analysis are deliberated. literature review provides a comprehensive foundation for understanding the key components of the project domain, informing the subsequent development and evaluation phases.

2.1 File Sharing System

The file-sharing system is crucial in the digital age, widely used in corporate, educational, and creative settings for efficient storage and collaboration. Online file-sharing services provide a platform where users can upload files and easily share content with others. The typical contents assumed in this context are media files, including images, photos, and short movies. A file-sharing system based on cloud storage is a service that allows several users to view the same collection of files stored in the cloud at the same time (Machida et al., 2022). Cloud storage is a common application of cloud computing where data is stored on multiple third-party servers instead of dedicated servers used in traditional networked data storage. When a user stores data in the cloud, they interact with a virtual server that creates the illusion of dedicated storage. The data server receives files from a client via the internet, records the data, and stores it. The client uses a web-based interface to visit the data server and retrieve the desired information. After that, the server either returns the files to the client or permits the client to read and modify the files stored on the actual server.

2.2 Multi-Factor Authentication

Multi-factor authentication (MFA) has emerged as an alternative way to improve security by requiring the user to provide more than one authentication factor, as opposed to only a password (Basin et al., 2018). Numerous organizations are embracing MFA to uphold user integrity within their networks, concurrently reducing the likelihood of unauthorized users infiltrating their systems. Authentication factors are usually of three kinds. Firstly, knowledge factors which are something the user knows such as passwords and life questions. Secondly, possessive factors are something the user has, such as a hardware token and a one-time password. Thirdly, inherence factors, which are something the users are, such as the fingerprint, iris, and voice.

One of the frequently encountered Multi-Factor Authentication (MFA) methods is the use of one-time passwords (OTPs)(Alsalem & Alshoshan, 2021). A One-Time Password (OTP) is a singular password string that holds validity for only one use and becomes immediately invalid after verification (Ali et al., 2020). The one-time passwords can be generated by security token, received through SMS and email, or dedicated by mobile applications such as Microsoft Authenticator and Google Authenticator.

2.3 Encryption

Data encryption involves the transformation of information (plaintext) into a coded form (ciphertext) to prevent unauthorized access. This process is achieved using algorithms and keys (Alenezi et al., 2020).

The primary objective of every encryption algorithm is to increase the complexity of the decryption process substantially, rendering it challenging without the assistance of the key employed during the encryption. Encryption algorithms are categorized into two main groups: Symmetric-key, also known as secret-key encryption, and Asymmetric-key, alternatively referred to as public-key encryption. Symmetric key encryption, a type of cryptosystem, involves utilizing the same key for encryption and decryption processes. This approach is also recognized as conventional encryption. A symmetric key in cryptography can be a numeric, alphanumeric, or special symbol. It plays a crucial role during both encryptions, applied to plain text, and decryption, applied to cipher text.

In 2001, the National Institute of Standards and Technology (NIST) recommended the Advanced Encryption Standard (AES) as the successor to DES. The AES algorithm is versatile, accommodating various combinations of data and key lengths, including 128 bits for both data and keys, 192 bits, and 256 bits. Depending on the chosen key length, the algorithm is denoted as AES-128, AES-192, or AES-256. Encryption involves a sequence of processing steps, comprising 10 rounds for 128-bit keys, 12 rounds for 192-bit keys, and 14 rounds for 256-bit keys. The security of an encryption algorithm is closely tied to the key size. Generally, larger key sizes provide stronger security. Hence, in theory, AES-256 offers a higher security margin than AES-192 or AES-128.

2.4 Hashing

The term hash function is a mathematical function designed to take any variable-sized input and generate a fixed-size output. Hash functions serve as a fundamental cryptographic building block utilized in numerous applications, including authentication, ensuring data integrity, and facilitating digital signatures. Examples of hash algorithms include Message Digest Algorithm 5 (MD5), MD4, HAVAL, FORT-256, SHA-family, and RIPEMD-family.

SHA-256, a cryptographic hash function, emerged in 2000 as a groundbreaking addition to the Secure Hash Algorithm (SHA) family. Its introduction marked a new era in cryptographic functions, and by 2002, it gained widespread recognition by being officially adopted as a Federal Information Processing Standards (FIPS) standard. SHA-256 is structured using the Merkle-Damgård construction and incorporates the Davis-Meyer mode. The compression function of SHA-256 involves 64 rounds, incorporating two distinct non-linear functions, cyclic rotations, and round-dependent constants. Notably, the resulting hash value generated by SHA-256 is 256 bits in length, contributing to its robustness in cryptographic applications. The SHA-512 provides a higher level of security compared to SHA-256 but it is generally slower than SHA-256 due to the larger size of the hash value. Combining these features enhances the algorithm's resistance to various attacks and ensures the integrity and security of the hashed data.

2.5 Comparison System of Existing System

This section will compare the existing and developed systems as shown in Table 1.

Table 1. System comparison

Attributes	Mega	Google Drive	MediaFire	Proposed System
Email verification for registration	YES (Email verification link)	NO	YES (Email verification link)	YES (OTP verification)
Strong password for registration	YES	NO	YES	YES
Multi-factor authentication for login	Enable by user in setting (OTP generated by authenticator app)	Enable by user in setting (OTP sent through phone)	NO	YES (OTP sent through email)
ReCAPTCHA when login and register	YES	NO	YES	YES
Secure password for the files shared link in the cloud	YES	NO	NO	YES

3. AUTHOR'S ARTWORK

The agile model is a system development approach emphasizing collaboration, flexibility, and continuous improvement through iterative and incremental cycles. It is based on the idea that requirements and solutions evolve. Therefore, it is better to deliver working software frequently in short cycles rather than trying to deliver a complete product at the end of a long development cycle. This project chose the agile model for the methodology because it is flexible and can adapt to change quickly. The Fig. 1 show the phase of agile model.

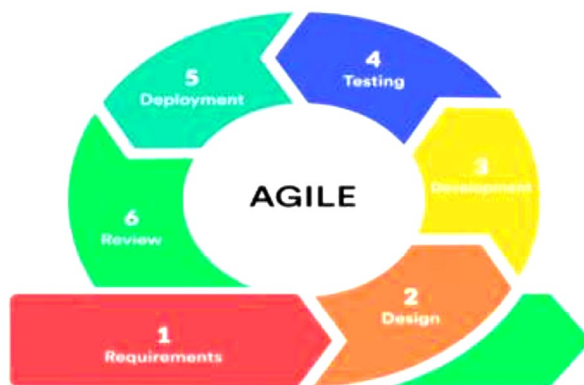


Fig. 1. Agile Model

3.1 System Development Workflow

This section provides a summary of the output of each phase, as shown in Table 2.

Table 2. System development workflow

Phase	Task	Output
Requirement phase	-Prepared requirement document	-Proposal -Gantt chart -Literature review
Design phase	-Design system interface -Design database -Design system architecture	-Wireframe -ERD diagram -System architecture diagram
Development phase	-Write system code -Create database	-Proposed system -System database
Testing phase	-Unit testing -Integrated testing -Security testing	-Bug fix -Unit testing result -Integrated testing result -Security testing result
Deployment phase	-Prepare user manual -Upload website to web server	-Readme.txt -Publish file-sharing system
Review phase	-User acceptance testing	-User Feedback through Google form
Requirement phase	-Prepared requirement document	-Proposal -Gantt chart -Literature review

3.2 System Analysis and Design

This section explores systematic analysis and design of the File Sharing System. The objective is to define system requirements and formulate a well-structured design. The system architecture is a conceptual framework that outlines the behavior of a system to meet its objectives. Figure 2 shows the system architecture design for the file-sharing system. This system utilizes a web-based interface allowing users to interact with it through their browser. Users are allowed to download, upload, encrypt, and share their files through the system. The system includes login and register modules and will store user data in the database. The server will store the files of the users.

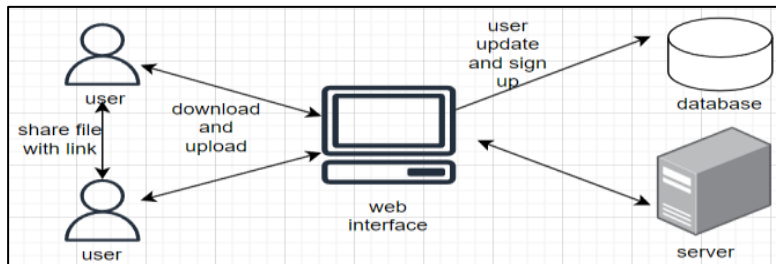


Fig. 2. System architecture diagram

Admins log in directly to the admin dashboard, while new users register via a one-time password before logging in. Both users and admins need a password and OTP for authentication. After login, users access the user dashboard, where they can create folders, upload files, open folders, download, delete, move, or share files. File sharing involves creating a link, setting a password, and sending the link. Admins have additional capabilities, including deleting user-uploaded files, allocating storage, and managing user accounts.

4. IMPLEMENTATION OF THE SYSTEM

This section discusses the implementation of security modules, which are strong passwords, one-time passwords, destroy sessions, and reCAPTCHA. Fig. 3 shows the strong password implementation in the proposed system. The password input of the users when registration will be compared to the regular expression to make sure the password validates with certain criteria. The system will require users to enter a password with a minimum 12-character length. The password should consist at least one number, one lowercase character, one uppercase letter and one special character.

```

password: /^(?=.*\d)(?=.*[a-zA-Z])[\da-zA-Z~!@#%&*_]{12,}
shareCode: /^[A-Za-z0-9]+$

const verify = (rule, value, reg, callback) => {
  if (value) {
    if (reg.test(value)) {
      callback()
    } else {
      callback(new Error(rule.message))
    }
  } else {
    callback()
  }
}

export default {
  email: (rule, value, callback) => {
    return verify(rule, value, regs.email, callback)
  },
  number: (rule, value, callback) => {
    return verify(rule, value, regs.number, callback)
  },
  password: (rule, value, callback) => {
    return verify(rule, value, regs.password, callback)
  },
  shareCode: (rule, value, callback) => {
    return verify(rule, value, regs.shareCode, callback)
  },
}

```

Fig. 3. Strong password implementation

Fig. 4 shows the code for sending the one-time password to the users on the proposed system. Before the users request a one-time password during registration, the code will check whether the email of the users exists in the database or not. The random string will be generated with five-character lengths and only consist of digits. After the email is sent, the previous one-time password associated with a specified email address will be disabled. The code set validation time for the one-time password is only 15 minutes.

```

public void sendEmailCode(String toEmail, Integer type) {
    // If it is registered, check if the email already exists
    if (type == Constants.ZERO) {
        UserInfo userInfo = userInfoMapper.selectByEmail(toEmail);
        if (null != userInfo) {
            throw new BusinessException("Email already exists");
        }
    }

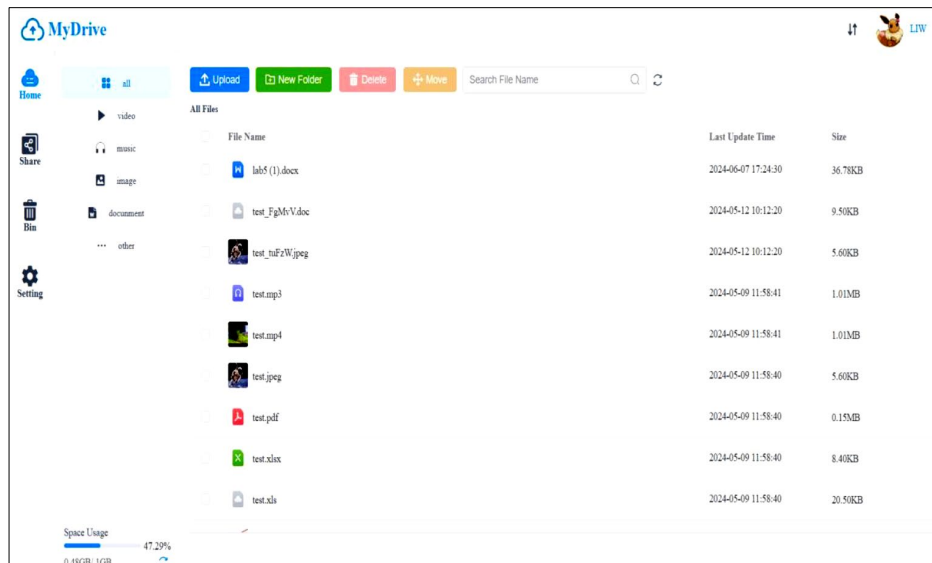
    String code = StringTools.getRandomNumber(Constants.LENGTH_5);
    sendEmailCode(toEmail, code);

    emailCodeMapper.disableEmailCode(toEmail);
    EmailCode emailCode = new EmailCode();
    emailCode.setCode(code);
    emailCode.setEmail(toEmail);
    emailCode.setStatus(Constants.ZERO);
    emailCode.setCreateTime(new Date());
    Calendar calendar = Calendar.getInstance();
    calendar.add(Calendar.MINUTE, amount: 15);
    emailCode.setExpiryTime(calendar.getTime());
    emailCodeMapper.insert(emailCode);
}

```

Fig. 4. OTP generation

Fig. 5 shows the home dashboard interface for the proposed system. The user can click on the upload button to upload the file, click on the new folder button to create a new folder, click on the delete button to delete selected files, and click on the share button to share the selected files. The dashboard will display the date modified for the folder and the size of the folder. The users can search the files through the file name. Besides, the files can be categorized by clicking on the video, image, document, music, and another button. The lower left corner of the dashboard will show the space usage of the users.



File Name	Last Update Time	Size
lab5 (1).docx	2024-06-07 17:24:30	36.78KB
test_FgMvV.doc	2024-05-12 10:12:20	9.50KB
test_nuEwJ.jpeg	2024-05-12 10:12:20	5.60KB
test.mp3	2024-05-09 11:58:41	1.01MB
test.mp4	2024-05-09 11:58:41	1.01MB
test.jpeg	2024-05-09 11:58:40	5.60KB
test.pdf	2024-05-09 11:58:40	0.15MB
test.xlsx	2024-05-09 11:58:40	8.40KB
test.xls	2024-05-09 11:58:40	20.50KB

Fig. 5. OTP generation

Fig. 6 shows the share record dashboard interface for the proposed system. The users are allowed to view the shared link record and copy the file shared link in this dashboard. The record will display the file name, expiration time for this shared link, and the number of views for this link. The users also can unshared the selected shared link.

<https://dx.doi.org/10.24191/jcrinn.v10i1.500>

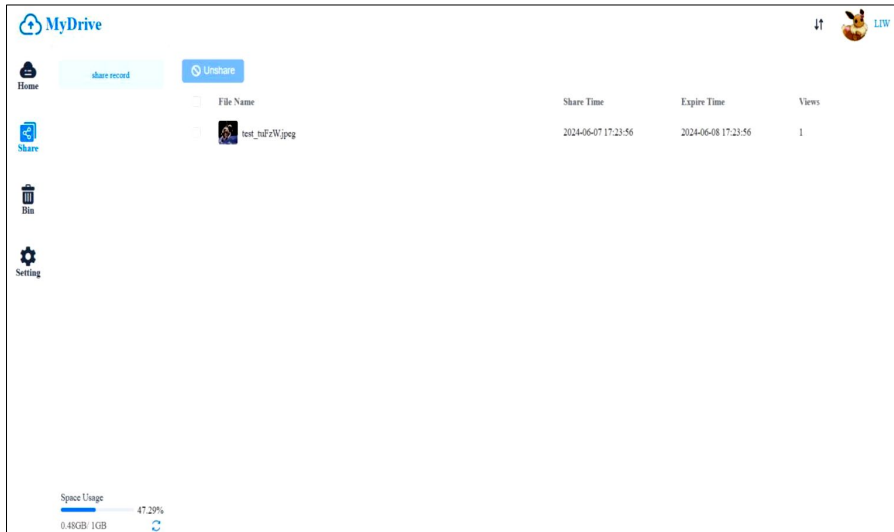


Fig. 6. Share Record Interface

5. RESULTS AND DISCUSSIONS

This section discusses the User Acceptance Test (UAT) form, which is used to evaluate test cases for the proposed system. The form applies the Likert scale from 1(Strong Satisfied) to 5(Strongly Dissatisfied). The section will show the result of the user acceptance test. The user acceptance form was created through the Google form and sent to the user. The Google form result was collected from 10 respondents, which are the students from the UTHM. All of the questions are agreed upon by the user, as shown in Table 3 below.

Table 3. UAT result

Test plan	Ranking				
	1	2	3	4	5
Users can receive the one-time password when registration and login					10
Display message easy to understand					10
Search the files					10
Upload files					10
Download files					10
Accept the file sharing link					10
Share the file with password					10
Unshared the file					10
Create folder				1	9
Rename folder				1	9
Move folder					10
Preview files				1	9
User friendly interface					10
System easy to use					10

The implementation of security modules and system properties ensures the system functionality is well-secured. The testing result also shows that the proposed system achieves the aim of the system and is accepted by the target user. The proposed system, a file-sharing system with strong password and one-time password authentication, has several advantages, which are allowing the user to upload files and store them in the system, sharing the files with other users, enforcing strong password requirements for user accounts, and enforce OTP authentication for login and registration.

6. CONCLUSION AND RECOMMENDATIONS

In conclusion, the secure file-sharing system addresses the critical need for a robust and user-friendly platform in today's digital landscape. Focused on fortifying existing systems, the project integrates advanced security measures, including multi-factor authentication which is strong passwords, and one-time passwords. The system fulfills its objectives by allowing users to upload and share files securely with strong password enforcement and one-time password authentication. However, it still has limitations, such as limited storage space, complexity in the login process due to multiple security layers, and the inability of users to encrypt stored files. Future implementations could focus on increasing storage capacity, simplifying the authentication process, and integrating encryption features to enhance security and user experience.

7. ACKNOWLEDGMENTS

The authors would like to thank the Faculty of Computer Science and Information Technology, University Tun Hussein Onn Malaysia for its support.

8. AUTHORS' CONTRIBUTIONS

Chee Lee Chong: Conceptualisation, methodology, formal analysis, investigation and writing-original draft; **Nur Ziadah Harun:** Conceptualisation, supervision, writing- review and editing, and validation.

9. REFERENCES

- Alenezi, M. N., Alabdulrazzaq, H., & Mohammad, N. Q. (2020). Symmetric encryption algorithms: Review and evaluation study. *International Journal of Communication Networks and Information Security*, 12(2), 256-272. <https://www.researchgate.net/publication/349324592>
- Ali, G., Dida, M. A., & Sam, A. E. (2020). Two-factor authentication scheme for mobile money: A review of threat models and countermeasures. *Future Internet*, 12(10), 1-27. <https://doi.org/10.3390/fi12100160>
- Alsaleem, B. O., & Alshoshan, A. I. (2021, March 27). Multi-factor authentication to systems login. In *IEEE 4th National Computing Colleges Conference* (pp. 1-4). IEEE Xplore. <https://doi.org/10.1109/NCCC49330.2021.9428806>
- Basin, D., Dreier, J., Hirschi, L., Radomirovic, S., Sasse, R., & Stettler, V. (2018). A formal analysis of 5G authentication. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security* (pp. 1383–1396). ACM. <https://doi.org/10.1145/3243734.3243846>

- Kulkarni, G., Waghmare, R., Palwe, R., Waykule, V., Bankar, H., & Koli, K. (2012). Cloud storage architecture. In the *7th International Conference on Telecommunication Systems, Services, and Applications (TSSA)* (pp. 76-81). IEEE Xplore. <https://doi.org/10.1109/TSSA.2012.6366026>
- Machida, F., Hasebe, K., Abe, H., & Kato, K. (2022). Analysis of Optimal File Placement for Energy-Efficient File-Sharing Cloud Storage System. *IEEE Transactions on Sustainable Computing*, 7(1), 75–86. <https://doi.org/10.1109/TSUSC.2020.3037260>



© 2025 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).