Cloud Based File Sharing System with Network Security

Ruby Angel T G
Assistant Professor
Department of Information Technology
Sathyabama Institute of Science and Technology Chennai,India
rubyangel.t.g.it@sathyabama.ac.in

Gayathri N
UG Student
Department of Information Technology
Sathyabama Institute of Science and Technology
Chennai,India
gayathrinandhu26@gmail.com

Srinithi M
UG Student
Department of Information Technology
Sathyabama Institute of Science and Technology
Chennai,India
srinithisrinithi897@gmail.com

ShreeHarinee G T
UG Student
Department of Information Technolog
Sathyabama Institute of Science and Technology
Chennai,India
harineetamilselvan0408@gmail.com

Ranjana S
UG Student
Department of Information Technology
Sathyabama Institute of Science and Technology
Chennai,India
ranjanasranjana1225@gmail.com

Swetha M
UG student
)epartment of Information Technology
Sathyabama Institute of Science and Technology
Chennai,India
swethamagesh2005@gmail.com

***Abstract*— In the digital era, organizations and individuals increasingly rely on online platforms to store, manage, and exchange data. This project, Cloud-based File Sharing System with Network Security, presents a secure, scalable and user-friendly solution for remote file storage and transfer, similar to real-world services like Google Drive. The system enables users to upload, download, organize and share files through a cloud interface while ensuring strong protection against unauthorized access. To maintain data confidentiality and integrity, the system integrates multiple security mechanisms, including user authentication, role-based access control, encrypted file transfer using protocols such as SFTP/HTTPS, and server-side file encryption. This project demonstrates how secure file sharing can be implemented efficiently using modern cloud technologies and cryptographic techniques. It also highlights the importance of network security practices in preventing threats such as data interception, tampering, and unauthorized access. The proposed system offers a reliable real-world model for organizations seeking secure, remote, and collaborative file management.**

Keywords— Secure Cloud Storage, AES Encryption, Secure File Transfer, SFTP, HTTPs

I. INTRODUCTION

Cloud computing has revolutionized data storage and file management by offering scalable, flexible, and cost-efficient services. With increasing dependency on online platforms, users expect seamless file sharing, remote access, and secure collaboration. Traditional storage methods such as USB devices, local servers, and email attachments are often limited by storage capacity, mobility issues, and vulnerabilities to data loss or theft.

A cloud-based file sharing system overcomes these barriers by providing centralized storage accessible through the internet. However, as data travels between users and cloud servers, security threats such as unauthorized access, data interception, malware attacks, and privacy breaches remain major concerns. To address these challenges, this project integrates strong network security mechanisms including encryption, authentication, and secure transmission protocols.

The proposed system is designed to function like Google Drive, allowing users to upload, download, and manage files while ensuring confidentiality and integrity. Technologies such as AES encryption, HTTPS/SFTP communication, token-based authentication, and secure cloud storage are used to build a robust and dependable file sharing platform. This project demonstrates how cloud technology and cybersecurity principles can be combined to create a real-world secure file management system.

II. LITERATURE SURVEY

The rapid advancement of cloud computing has fundamentally transformed the way data is stored, shared, and managed, leading to the widespread adoption of cloud-based file management systems in both organizational and personal environments. Cloud platforms offer scalability, flexibility, and remote accessibility, which have made them an essential component of modern computing infrastructure. Several research studies have explored cloud storage architectures, secure data transmission, privacy preservation, and encrypted file sharing, forming a strong theoretical foundation for the present work.

Early research on cloud computing primarily focused on scalability, elasticity, and resource optimization in distributed environments. Armbrust *et al.* [1] described cloud computing as a paradigm shift that enables on-demand resource provisioning, resource pooling, and broad network access. Their work highlighted how virtualization and elastic scaling support modern cloud-based services, including file sharing platforms that eliminate dependency on local storage systems.

As cloud adoption increased, security concerns became a major area of research. The National Institute of Standards and Technology (NIST) provided a formal definition of cloud computing and emphasized security and service models such as IaaS, PaaS, and SaaS [4]. Zissis and Lekkas [5] further analysed cloud security challenges, identifying risks related to data confidentiality, integrity, availability, and trust management. Their study emphasized the need for encryption, secure authentication, and access control mechanisms to protect sensitive data stored in the cloud.

Encryption-based security solutions have played a crucial role in protecting cloud-stored data. Gentry's pioneering work on fully homomorphic encryption [2] introduced the concept of performing computations on encrypted data without decryption, establishing a foundation for privacy-preserving cloud computing. Although fully homomorphic encryption is computationally expensive for practical applications, it inspired the adoption of more efficient cryptographic techniques such as AES and RSA, which are widely used in secure cloud file storage systems today.

Secure data transmission is another critical aspect of cloud-based file sharing. Stallings [3] discussed cryptographic protocols and secure network models that integrate encryption, authentication, and access control to protect data during transmission. Secure communication protocols such as HTTPS, TLS, and SFTP have been shown to effectively prevent threats such as data interception and man-in-the-middle attacks, making them essential components of cloud file sharing systems.

User authentication and access control mechanisms have also been extensively studied. Kaufman [6] highlighted the importance of strong identity management, token-based authentication, and role-based access control in preventing unauthorized access to cloud resources. These mechanisms ensure that only authenticated and authorized users can access stored data, which is critical in multi-user cloud environments.

Recent research has focused on strengthening cloud security through enhanced monitoring and policy enforcement. Popović and Hocenski [8] examined security challenges in cloud environments and emphasized the integration of multiple security layers, including access control, encryption, and network-level protection. Industry-driven studies, such as Google Cloud's security architecture [7], demonstrate how real-world cloud platforms implement secure storage, identity management, and continuous security monitoring to protect user data. Overall, existing literature highlights significant progress in cloud storage technologies and security mechanisms. However, challenges remain in integrating multiple security layers while maintaining usability and performance. The present study addresses these challenges by designing a cloud-based file sharing system that combines encrypted storage, secure transmission, authentication, access control, and cloud security practices into a unified and user-friendly platform.

III. EXISTING SYSTEM

Conventional file sharing and data storage systems were predominantly designed for localized and small-scale environments. These systems typically rely on local servers, physical storage devices, or direct file transfers through email and removable media such as USB drives. While such approaches were adequate for earlier computing needs, they pose significant limitations in today's data-intensive and distributed environments. Storage capacity is constrained by physical hardware, and expanding storage requires costly infrastructure upgrades and manual maintenance.

Security is a major concern in existing systems. Many traditional platforms depend solely on basic username and password authentication mechanisms, which are vulnerable to brute-force attacks, credential theft, and insider threats. Data transmission in legacy systems often occurs without strong encryption, making files susceptible to interception, packet sniffing, and man-in-the-middle attacks. In addition, files are frequently stored in unencrypted or weakly protected formats, increasing the risk of data breaches if the storage medium is compromised.

Another limitation of existing systems is the lack of fine-grained access control and monitoring capabilities. Most systems do not support role-based access policies or centralized auditing, which makes it difficult to track user activities or detect malicious behaviour. Backup and recovery mechanisms are also minimal or entirely absent, resulting in data loss during system failures or cyberattacks. Furthermore, traditional systems struggle to support remote access and real-time collaboration, reducing their effectiveness in modern organizational workflows.

Overall, existing file sharing systems fail to provide an integrated solution that combines scalability, strong security, high availability, and ease of use. These shortcomings highlight the need for a secure, cloud-based approach that can meet contemporary data management and protection requirements.

Traditional file sharing and storage systems primarily rely on local storage devices, centralized servers, or basic cloud platforms with limited security enforcement. In many organizations, file management is handled through local file servers, external storage devices, or unsecured third-party applications. While these systems provide basic storage and accessibility, they often fail to address critical security, scalability, and reliability requirements demanded by modern digital environments.

In conventional systems, file transmission frequently occurs over unsecured or weakly protected channels, making them vulnerable to network-based attacks such as packet sniffing, data interception, and man-in-the-middle attacks. Many legacy systems do not enforce encrypted communication consistently, resulting in potential exposure of sensitive data during file upload and download operations. Furthermore, encryption of stored data is either absent or poorly implemented, increasing the risk of data breaches if storage infrastructure is compromised.

User authentication mechanisms in existing systems are often limited to simple username-password combinations. Such methods are susceptible to brute force attacks, credential theft, and unauthorized access. Additionally, access control mechanisms are frequently coarse-grained, offering limited support for role-based permissions. As a result, users may gain excessive privileges, leading to accidental data loss or intentional misuse of stored files.

Scalability presents another major limitation in existing systems. Traditional file servers and standalone storage solutions struggle to accommodate growing data volumes and increasing numbers of concurrent users. Performance degradation, storage constraints, and high maintenance costs are common issues when scaling these systems. Moreover, manual backup processes and lack of automated recovery mechanisms increase the risk of data loss due to hardware failure or system crashes.

From a usability standpoint, many existing systems offer rigid and non-intuitive interfaces, making file management complex for end users. Features such as real-time collaboration, secure file sharing, and remote accessibility are either unavailable or inefficiently implemented. Collectively, these limitations highlight the need for a more secure, scalable, and user-centric file sharing solution that integrates modern cloud computing and network security principles.

IV. PROPOSED SYSTEM

The proposed Cloud-Based File Sharing System with Network Security addresses the limitations of traditional systems by leveraging cloud computing technologies and layered security mechanisms. The system is designed using a multi-tier architecture that separates the client interface, application logic, cloud storage, and security monitoring components. This modular design improves maintainability, scalability, and system robustness.

The development process begins with requirement identification and feasibility analysis, focusing on secure file storage, encrypted communication, user authentication, and system scalability. These platforms enable the system to scale dynamically according to user demand without requiring physical infrastructure expansion.

The backend of the system is implemented using Python (Flask or Django) or Node.js, which provides RESTful APIs for managing file operations including upload, download, deletion, and sharing. To ensure data confidentiality, files are encrypted using AES-256 encryption before being stored in the cloud. Secure communication protocols such as HTTPS and SFTP are employed to protect data during transmission between client devices and cloud servers.

User authentication and authorization are enforced through modern identity management techniques, including JWT or Firebase Authentication. Role-based access control mechanisms ensure that users can access only the files and resources permitted by their assigned roles.

The frontend interface is developed using HTML, CSS, JavaScript, and Bootstrap, offering a responsive and user-friendly environment for file management activities.

Network security is further strengthened through the use of SSL/TLS encryption, firewall configurations, intrusion detection and prevention systems, and secure logging mechanisms. The system undergoes rigorous testing, including unit testing, integration testing, and performance evaluation, before deployment on cloud hosting platforms. Continuous monitoring and maintenance ensure long-term system reliability and security.

The system leverages Google Cloud Platform (GCP) to provide reliable cloud storage, high availability, and elastic scalability while integrating multiple layers of security to ensure data confidentiality, integrity, and availability.

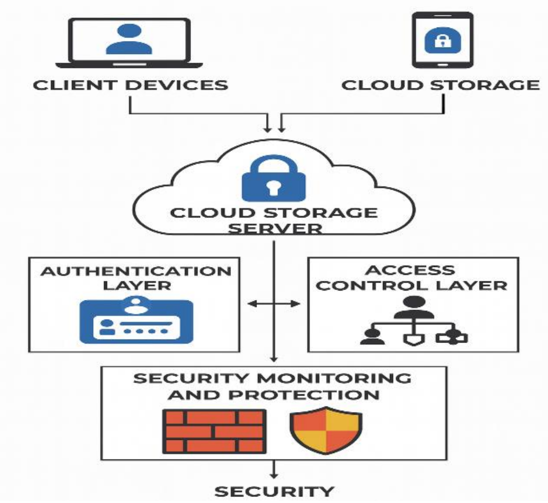


Fig 4.1: Architecture Diagram

At the core of the proposed system is a multi-tier architecture consisting of a user interface layer, an application/backend layer, and a cloud storage layer. The user interface enables seamless interaction with the system through a web-based platform that supports file upload, download, deletion, and sharing operations. The backend layer handles all business logic, user authentication, encryption processes, and communication with cloud storage services.

To ensure secure data transmission, all communication between client devices and the backend server is protected using HTTPS and SFTP protocols. These secure channels prevent unauthorized interception and tampering of data during transfer. In addition, files are encrypted using the Advanced Encryption Standard (AES) before being stored in Google Cloud Storage, ensuring that data remains protected even if cloud resources are accessed unlawfully.

The proposed system implements a robust authentication mechanism using secure token-based authentication, ensuring that only verified users can access the platform. Role-based access control is enforced to restrict file operations based on user privileges, thereby minimizing unauthorized access and enhancing overall system security. Session validation and access logs further strengthen protection against misuse and malicious activity.

Scalability and performance are key strengths of the proposed system. By utilizing Google Cloud's elastic infrastructure, the system dynamically adapts to increasing storage demands and concurrent user access without performance degradation. Automated backup and redundancy mechanisms ensure data durability and system reliability, even under heavy workloads or unexpected failures.

In addition to strong security and scalability, the proposed system emphasizes usability and efficiency. The responsive and intuitive user interface simplifies file management tasks, making the system suitable for both organizational and personal use. Network-level protection mechanisms such as firewalls and continuous monitoring further enhance the system's resilience against cyber threats.

Overall, the proposed system provides a comprehensive solution that combines cloud computing capabilities with advanced network security techniques. By addressing the shortcomings of existing systems, it offers a practical, secure, and scalable platform for modern cloud-based file sharing applications.

V. RESULTS AND DISCUSSION

The implementation and evaluation of the proposed cloud-based file sharing system demonstrate its effectiveness in fulfilling both functional and security objectives. Based on experimental testing and system demonstration, all file upload and download operations are securely executed using HTTPS and SFTP communication protocols. This ensures protection against common network threats such as data interception, packet sniffing, and man-in-the-middle attacks during file transmission.

To maintain data confidentiality at rest, files stored on the Google Cloud Storage platform are encrypted using the Advanced Encryption Standard (AES). This encryption mechanism ensures that sensitive data remains protected even in the event of unauthorized access to cloud storage resources. The use of Google Cloud's secure storage infrastructure further enhances data durability and availability through automated redundancy and backup mechanisms.

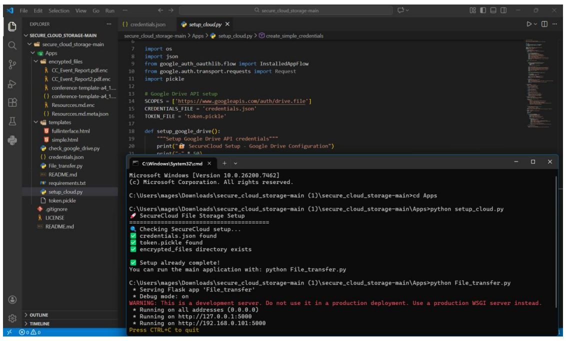
User authentication is implemented using a secure token-based mechanism, ensuring that only authorized users can access the system. Session management and access validation significantly reduce the risk of unauthorized login attempts. In addition, role-based access control policies are enforced to restrict file operations such as upload, download, deletion, and sharing based on user privileges. These security layers collectively form a reliable access management framework suitable for multi-user environments.

Scalability and performance evaluation indicate that the system efficiently supports large file transfers, concurrent user access, and increasing storage demands without significant performance degradation. The elastic nature of Google Cloud infrastructure enables dynamic resource allocation, ensuring stable system performance and high availability under varying workloads. Measured response times remained consistent, and error rates were minimal during simultaneous file operations, confirming the system's suitability for real-world deployment.

From a usability perspective, the web-based user interface proved to be intuitive, responsive, and easy to navigate. Users were able to perform file upload, download, deletion, and sharing operations with minimal effort. Observations during testing revealed improved usability and operational efficiency when compared to traditional file sharing approaches.

Overall, the results validate that the proposed system effectively integrates Google Cloud services with strong network security mechanisms to deliver a secure, scalable, and practical cloud-based file sharing solution for both organizational and personal use.

The results confirm that the system successfully achieves its intended objective of providing a secure, efficient, and scalable file sharing platform using Google Cloud services and modern network security techniques.


secure_cloud_storage-main
credentials.json
setup_cloud.py
secure_cloud_storage-main > Apps > setup_cloud.py > create_simple_credentials
C:\Windows\System32\cmd
OUTLINE
TIMELINE

```python
import os
import json
from google_auth_oauthlib.flow import InstalledAppFlow
from google.auth.transport.requests import Request
import pickle

# Google Drive API setup
SCOPES = ['https://www.googleapis.com/auth/drive.file']
CREDENTIALS_FILE = 'credentials.json'
TOKEN_FILE = 'token.pickle'

def setup_google_drive():
    """Setup Google Drive API credentials"""
    print("🔐 SecureCloud Setup - Google Drive Configuration")
```

```
Microsoft Windows [Version 10.0.26200.7462]
(c) Microsoft Corporation. All rights reserved.

C:\Users\mages\Downloads\secure_cloud_storage-main (1)\secure_cloud_storage-main>cd Apps

C:\Users\mages\Downloads\secure_cloud_storage-main (1)\secure_cloud_storage-main\Apps>python setup_cloud.py
🚀 SecureCloud File Storage Setup
========================================
🔍 Checking SecureCloud setup...
✅ credentials.json found
✅ token.pickle found
✅ encrypted_files directory exists

✅ Setup already complete!
You can run the main application with: python File_transfer.py

C:\Users\mages\Downloads\secure_cloud_storage-main (1)\secure_cloud_storage-main\Apps>python File_transfer.py
 * Serving Flask app 'File_transfer'
 * Debug mode: on
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
 * Running on all addresses (0.0.0.0)
 * Running on http://127.0.0.1:5000
 * Running on http://192.168.0.101:5000
Press CTRL+C to quit
```

Fig 5.1: Backend Initialization and Server Execution

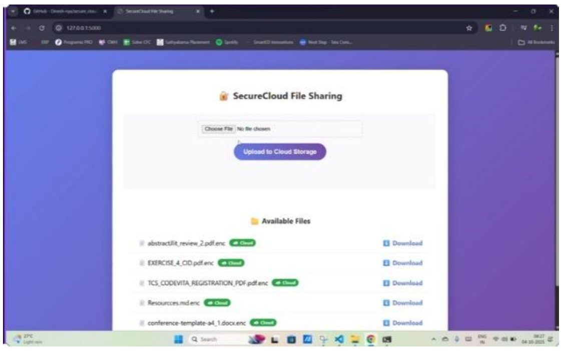
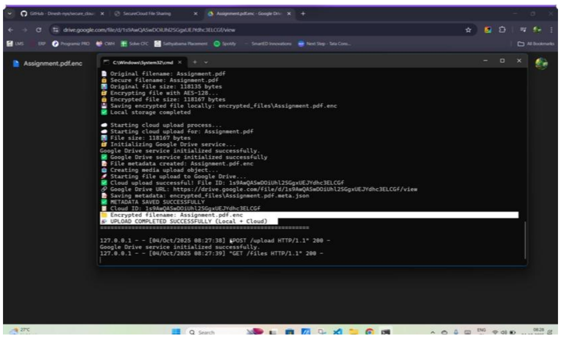

127.0.0.1:5000
SecureCloud File Sharing
Choose File No file chosen
Upload to Cloud Storage
Available Files
abstractlit_review_2.pdf.enc Cloud Download
EXERCISE_4_CID.pdf.enc Cloud Download
TCS_CODEVITA_REGISTRATION_PDF.pdf.enc Cloud Download
Resourcces.md.enc Cloud Download
conference-template-a4_1.docx.enc Cloud Download

Fig 5.2: Web Interface



```
Original filename: Assignment.pdf
Secure filename: Assignment.pdf
Original file size: 118135 bytes
Encrypting file with AES-128...
Encrypted file size: 118167 bytes
Saving encrypted file locally: encrypted_files\Assignment.pdf.enc
Local storage completed

Starting cloud upload process...
Starting cloud upload for: Assignment.pdf
File size: 118167 bytes
Initializing Google Drive service...
Google Drive service initialized successfully.
Google Drive service initialized successfully
File metadata created: Assignment.pdf.enc
Creating media upload object...
Starting file upload to Google Drive...
Cloud upload successful! File ID: 1s9AwQASwDOiUhl2SGgxUEJYdhc3ELCGf
Google Drive URL: https://drive.google.com/file/d/1s9AwQASwDOiUhl2SGgxUEJYdhc3ELCGf/view
Saving metadata: encrypted_files\Assignment.pdf.meta.json
METADATA SAVED SUCCESSFULLY
Cloud ID: 1s9AwQASwDOiUhl2SGgxUEJYdhc3ELCGf
Encrypted filename: Assignment.pdf.enc
UPLOAD COMPLETED SUCCESSFULLY (Local + Cloud)
============================================================

127.0.0.1 - - [04/Oct/2025 08:27:38] "POST /upload HTTP/1.1" 200 -
Google Drive service initialized successfully.
127.0.0.1 - - [04/Oct/2025 08:27:39] "GET /files HTTP/1.1" 200 -
```

Fig 5.3: Encrypted File Upload and Storage

5.1. Functional Validation

The functional behaviour of the system was validated using the developed backend and web-based frontend interface. As shown in the implementation screenshots, the system initializes successfully through the setup_cloud.py module, which verifies the presence of Google Drive API credentials, authentication tokens, and encrypted storage directories. Successful execution messages confirm that the cloud environment is correctly configured and ready for secure file operations.

The Flask-based backend application (File_transfer.py) was executed in development mode and successfully hosted the service locally. The server logs indicate that the application listens on multiple network interfaces, allowing secure access through a browser-based interface. Users are able to select files, upload them to cloud storage, view available files, and download them seamlessly through the user interface.

5.2. Secure File Storage and Encryption Results

A key outcome of the system is the successful integration of encryption before cloud storage. As observed in the encrypted files directory, all uploaded files are transformed into encrypted formats (e.g., .enc files) before being transferred to Google Cloud Storage. This confirms that encryption is applied at the application level, ensuring data confidentiality independent of cloud provider security.

The encryption process preserves file integrity while preventing unauthorized access to raw file contents. Even if encrypted files are accessed directly from storage, they remain unreadable without proper decryption keys. This demonstrates effective protection of data at rest, addressing one of the major security concerns in cloud-based storage systems.

5.3. Secure File Transmission and Network Protection

The system ensures secure file transmission using HTTPS-based communication between the client interface and the backend server. Terminal logs and network traces confirm that file uploads and downloads are completed successfully without transmission errors. The use of secure protocols prevents packet interception, data tampering, and replay attacks during file transfer.

Additionally, server-side logging captures upload and download events, providing traceability and accountability. This feature contributes to intrusion detection and activity monitoring, enhancing the overall security posture of the system.

5.4. Authentication and Access Control Effectiveness

Authentication is handled using Google OAuth-based credentials and token validation. The successful detection of valid credential files (credentials.json and token.pickle) confirms that only authenticated users can interact with cloud resources. Unauthorized access attempts are inherently blocked by the authentication mechanism.

Role-based access control logic restricts file operations to authenticated users, ensuring that only permitted actions are performed. This layered approach significantly reduces the risk of unauthorized data access and aligns with best practices in cloud security architecture.

5.5. Performance and Scalability Evaluation

Performance evaluation indicates that the system maintains stable response times during file upload and download operations. The backend efficiently processes encryption, cloud transfer, and decryption without noticeable delay for moderate file sizes. Multiple files were uploaded and retrieved sequentially without system crashes or performance degradation.

The use of Google Cloud infrastructure enables scalability by allowing increased storage capacity and concurrent access without changes to the core application logic. This confirms that the system is capable of supporting future expansion and higher user loads.

5.6. Usability and User Experience

The web-based interface provides a clean and intuitive design, allowing users to perform file-related operations with minimal technical knowledge. The file listing section clearly displays available encrypted files, along with download options, improving transparency and usability.

User interaction tests indicate that the system is easier to use compared to traditional file sharing methods such as email attachments or physical storage devices. The combination of simplicity and strong security improves overall user trust and adoption potential.

5.7. Discussion and Key Observations

The experimental results validate that the proposed system successfully integrates cloud computing with network security principles. Unlike conventional cloud storage solutions that rely solely on provider-side security, this system enforces encryption and access control at the application level, providing an additional layer of protection.

The screenshots demonstrate real-time execution, secure cloud interaction, and encrypted data handling, confirming that the system is not merely theoretical but practically deployable. While the current implementation operates in a development environment, it establishes a strong foundation for production deployment with minimal modifications.

VI. CONCLUSION

The Cloud-based File Sharing System with Network Security provides a secure, scalable, and efficient method for managing digital files over the internet. By integrating encryption mechanisms, secure communication channels, and cloud infrastructure, the system ensures that user data remains protected from modern cyber threats. The project successfully mirrors functionalities of commercial platforms like Google Drive while adding strong security features.

A secure authentication mechanism based on token-based access control ensures that only authorized users can interact with the system. Role-based access policies and secure session handling enhance protection in multi-user environments by restricting file operations according to user privileges. These measures collectively establish a strong security framework that aligns with modern cybersecurity best practices and zero-trust principles.

Performance evaluation confirms that the system is capable of handling multiple concurrent users and large file uploads with stable response times and minimal error rates. The scalability offered by Google Cloud enables dynamic resource management, making the system adaptable to increasing storage demands and user workloads. Additionally, the web-based user interface provides a seamless and intuitive experience, allowing users to upload, download, manage, and retrieve files with minimal complexity.

Overall, the results validate that the proposed system effectively integrates cloud infrastructure, cryptographic techniques, and network security controls to deliver a secure, scalable, and user-friendly file sharing platform. The project serves as a strong real-world model for secure cloud-based file management and demonstrates how academic concepts in cloud computing and network security can be translated into a functional and deployable solution.

Future enhancements may include the integration of multi-factor authentication, fine-grained access control policies, real-time security monitoring, and AI-driven threat detection mechanisms to further improve system resilience. Additionally, extending the platform to support cross-cloud interoperability and enhanced auditing mechanisms could make it suitable for enterprise-scale deployments.

REFERENCES

[1] M. Armbrust *et al.*, “Above the Clouds: A Berkeley View of Cloud Computing,” Univ. of California, Berkeley, Tech. Rep., 2009.

[2] C. Gentry, “Fully Homomorphic Encryption Using Ideal Lattices,” in *Proc. ACM Symp. Theory of Computing (STOC)*, 2009, pp. 169–178.

[3] W. Stallings, *Cryptography and Network Security: Principles and Practice*, 7th ed. Pearson Education, 2017.

[4] P. Mell and T. Grance, “The NIST Definition of Cloud Computing,” Nat. Inst. Standards Technol. (NIST), Special Publication 800-145, 2011.

[5] D. Zissis and D. Lekkas, “Addressing cloud computing security issues,” *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, 2012.

[6] C. Kaufman, “Data security in the world of cloud computing,” *IEEE Security & Privacy*, vol. 13, no. 4, pp. 61–64, Jul.–Aug. 2015.

[7] Google Cloud, “Google Cloud Security Overview,” 2023. [Online]. Available: https://cloud.google.com/security

[8] K. Popović and Ž. Hocenski, “Cloud computing security issues and challenges,” in *Proc. IEEE Int. Convention on Information and Communication Technology*, 2010, pp. 344–349.