Journal of **N**onlinear **A**nalysis and **O**ptimization
Vol. 16, Issue. 1: 2025
ISSN : **1906-9685**

A Network Protocol That Uses Secure Shell for Secure, Encrypted File Transfers, Offering an Alternative to the Traditional File Transfer Protocol

[1] **U.Indu Sekhar** , [2] Maguluri Ramadevi, [3] Boggavarapu Venkata Sai, [4] Mundruavinash

1Asst.Professor, Department of CSE-Cyber Security
2,3,4, UG Scholar, Department of CSE-Cyber Security
Chalapathi Institute of Technology, Guntur, Andhra Pradesh, India-522016.

ABSTRACT

The Secure File Transfer Protocol (SFTP) is a robust and reliable protocol designed for secure file transfers over a network, leveraging the Secure Shell (SSH) protocol for encryption and authentication. SFTP addresses the inherent vulnerabilities of traditional file transfer protocols, such as FTP, by providing a secure mechanism that ensures data confidentiality, integrity, and authentication. It facilitates encrypted communication, preventing unauthorized access and data breaches during transmission [4]. SFTP operates by establishing a secure channel between a client and a server, where all data exchanged—whether file contents, commands, or authentication details—is encrypted. Authentication mechanisms include passwords, SSH keys, and optional two-factor authentication, enhancing access control [2]. Unlike older protocols, SFTP encrypts the entire session, protecting not just the files but also the metadata and communication commands.

This protocol supports various operations, such as uploading, downloading, renaming, and deleting files, as well as directory management. It also enables features like resuming interrupted transfers, ensuring reliable and efficient data exchange [1]. The cross-platform compatibility of SFTP allows it to function seamlessly across diverse operating systems, making it versatile for various applications. Commonly used in industries like finance, healthcare, and government, SFTP helps organizations comply with regulatory standards such as GDPR, HIPAA, and PCI DSS by ensuring secure data transmission [5]. Its advantages over alternatives like FTP, FTPS, and SCP include stronger encryption, ease of use, and a more comprehensive set of features for file management. In conclusion, SFTP is an indispensable tool for secure file management, offering a high level of security and operational efficiency [6]. Its widespread adoption underscores its critical role in safeguarding sensitive data in today's digital landscape.
Keywords: Cross-Platform, Secure File Transfer Protocol, Secure Shell (SSH), and Secure Algorithm.

1. Introduction

The SFTP Project focuses on implementing a secure, efficient, and reliable system for transferring files over a network using the Secure File Transfer Protocol (SFTP). The project aims to address the growing need for data security in file transfer processes, ensuring that sensitive information is protected from unauthorized access, interception, and tampering [8]. Secure Data Transmission: Ensure all file transfers are encrypted, protecting the confidentiality and integrity of the data. User Authentication: Implement robust authentication mechanisms, including password-based, SSH key-based, and two-factor authentication, to restrict access to authorized users only. Ease of Use: Develop a user-friendly interface or system for managing file transfers effectively. Compliance: Adhere to regulatory standards, such as HIPAA, GDPR, or PCI DSS, for secure data handling. Scalability and Compatibility: Design the system to function across various platforms and handle growing data transfer needs. Encrypted File Transfers: Use SSH-based encryption to secure data during transit [10]. File Management Operations: Support functionalities like uploading, downloading, renaming, deleting, and resuming interrupted transfers. Access Control: Define and manage user roles, permissions, and access levels. Logging and Monitoring: Maintain logs of all transfer activities for auditing and troubleshooting. Automated Transfers: Include options for scheduling and automating repetitive file transfer tasks. Cross-Platform Support: Ensure compatibility with major operating systems [11].

2. EXISTING SYSTEM

The existing systems for Secure File Transfer Protocol (SFTP) typically consist of a client-server architecture where the SFTP server handles file storage, security, and user access, while the SFTP client allows users to securely transfer files over a network [15]. These systems are widely used for secure data exchange in environments such as healthcare, finance, and business.

2.1. Key components of the existing SFTP systems include: SFTP Server: Hosts files and manages user authentication and secure connections. Popular implementations include Open SSH and File Zilla Server. SFTP Client: Provides a user interface (CLI or GUI) for users to interact with the SFTP server. Common clients are WinSCP, Cyber duck, and File Zilla. Encryption: Uses strong encryption (like AES) to secure data during transfer. Authentication: Employs password-based or SSH key-based authentication, and optionally multi-factor authentication (MFA) for added security. File Management: Allows operations like uploading, downloading, renaming, and deleting files, as well as directory management [14]. While existing SFTP systems offer strong security and reliability, challenges such as key management, user experience, and performance optimization still exist. The current systems are also evolving to integrate with cloud platforms and automate file transfer processes.

2.2. Performance Analysis: File Transfer Speed: The system should optimize for large file transfers by minimizing overhead introduced by encryption. Techniques like compression before transfer or parallel transfers can improve performance [11]. Connection Management: The server should be capable of handling multiple simultaneous connections without significant performance degradation. Resumption of Interrupted Transfers: Implementing file transfer resumption ensures that users can continue transfers after a connection drop, saving time and resources.

3. PROPOSED SYSTEM

The proposed system for the Secure File Transfer Protocol (SFTP) project aims to enhance the security, usability, and performance of file transfers while addressing limitations found in existing systems. The goal is to create a more secure, scalable, and user-friendly solution for organizations and individuals to securely transfer files over a network [3]. Key Objectives of the Proposed System: Enhanced Security: Ensure end-to-end encryption for all data transfers, improving data confidentiality and integrity. Robust Authentication: Implement advanced authentication mechanisms such as multi-factor authentication (MFA) and SSH key rotation to mitigate the risks of unauthorized access. Improved User Experience: Develop a user-friendly interface for both technical and non-technical users, allowing seamless management of files and directories. Optimized Performance: Enhance the transfer speed and efficiency, especially for large files, by implementing features like compression, parallel file transfers, and resumption of interrupted transfers [10].

Automation and Scheduling: Integrate features for automated file transfers, scheduled uploads/downloads, and regular backups to improve system efficiency. Cloud Integration: Enable secure file transfers between on-premises systems and cloud storage services (e.g., AWS S3, Google Cloud Storage) to support hybrid IT environments. Audit and Monitoring: Implement comprehensive logging and real-time monitoring to track user activity, file transfer status, and security events for auditing and compliance purposes.

4. SYSTEMSTUDY

A system study on secure file transfer protocols involves analyzing, evaluating, and selecting the best-suited protocols to ensure the secure exchange of files in your project. The study typically focuses on: Understanding Requirements: Identify your project's needs, such as file types, volume, frequency, speed, compliance requirements (e.g., GDPR, HIPAA), and the sensitivity of data being transferred. Evaluating Protocols: Explore secure file transfer protocols like: SFTP (Secure File Transfer Protocol): Uses SSH for secure authentication and encrypted data transfer. FTPS (FTP Secure): Adds SSL/TLS encryption to the standard FTP protocol. HTTPS: Securely transfers files via HTTP with SSL/TLS encryption. AS2 (Applicability Statement 2): Used for business-to-business file transfers with encryption and digital signatures. SCP (Secure Copy Protocol): A simplified option for transferring files over SSH. Assessing Security Features: Look into encryption standards, authentication mechanisms, and protections against attacks like man-in-the-middle (MitM) or data breaches. Compatibility and Integration: Ensure the protocol integrates well with existing systems, tools, and workflows in your project. Performance: Consider the speed, reliability, and efficiency of file transfers. Compliance and Governance: Check if the protocol aligns with industry-specific regulatory requirements. Cost and Scalability: Assess implementation costs and scalability for future growth. The result of the study will guide you in choosing a protocol that provides the right balance of security, functionality, and efficiency for your project's file transfer needs.

4.1 DESIGN AND ARCHITECTURE

The architecture of a **secure file transfer protocol** involves several components and design principles that ensure the confidentiality, integrity, and authenticity of data during transfer. Below is an overview of the architecture:

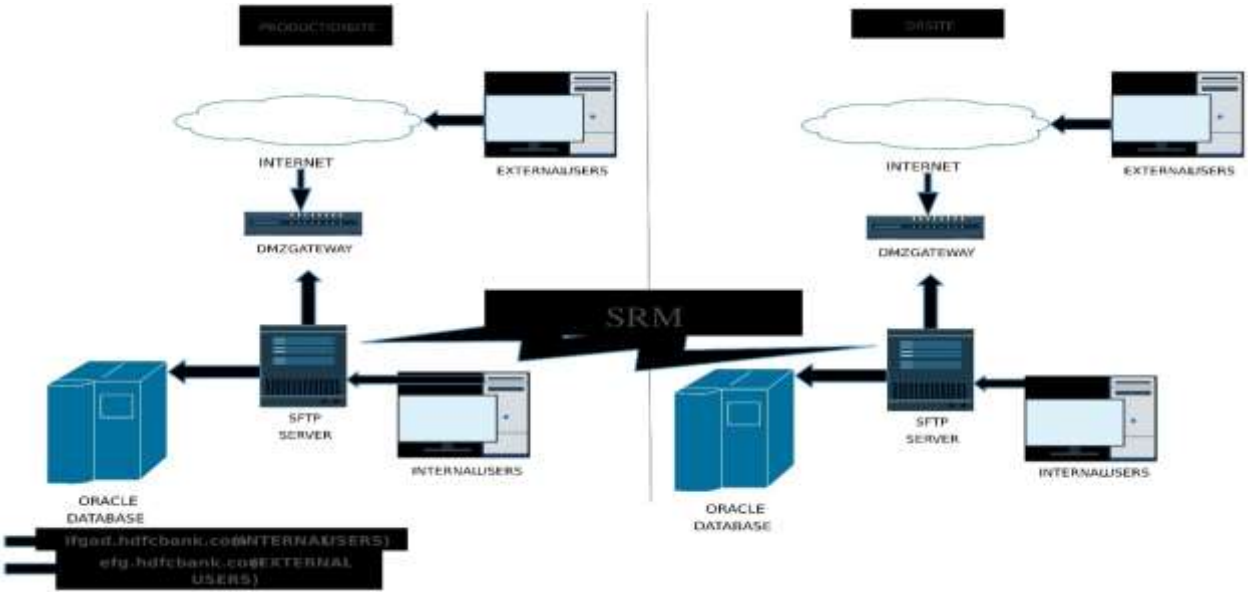

Fig: 1.System Design

Core Components: Client: The entity initiating the file transfer. Provides a user interface or APIs for sending and receiving files. Handles encryption and authentication before transmission. Server : The entity that receives and processes file transfer requests. Validates the client's credentials and enforces security policies. Stores or forwards files securely. Communication Channel: A secure channel that ensures encrypted transmission [5]. Examples are TLS for HTTPS-based transfers, or SSH for SFTP. Encryption Mechanisms :Data-at-Rest Encryption: Protects files stored on the server. Data-in-Transit Encryption: Protects files during transfer (e.g., AES, RSA). Authentication and Authorization Ensures only authorized users can access files [7]. Methods: Username/password, public/private key pairs, multi-factor authentication (MFA). File Integrity Validation uses checksums or hash functions (e.g., SHA-256) to verify that files have not been tampered with during transfer.

5. CONCLUSION

The Secure File Transfer Protocol (SFTP) is an essential solution for securely transferring files across networks, ensuring data confidentiality, integrity, and secure access. By leveraging SSH (Secure Shell), SFTP provides robust encryption, strong authentication, and file integrity checks, making it a reliable and widely used protocol in industries that handle sensitive or critical data. SFTP addresses the growing need for secure communication, enabling organizations to protect their data from unauthorized access, cyber attacks, and potential breaches. Its compatibility with various

platforms, automation capabilities, and compliance with security standards like GDPR, HIPAA, and PCI DSS make it an ideal choice for secure file transfer operations in both small-scale and enterprise environments. By implementing SFTP alongside best practices such as key-based authentication, regular audits, and access control measures, organizations can ensure the secure transfer of their files and safeguard their digital assets in today's security-conscious landscape. As data security challenges continue to grow, adopting secure protocols like SFTP is a fundamental step toward building a resilient and trustworthy IT infrastructure.

REFERENCES

[1] Kalyankumar Dasari, Mohmad Ahmed Ali, NB Shankara, K Deepthi Reddy, M Bhavsingh, K Samunnisa, "A Novel IoT-Driven Model for Real-Time Urban Wildlife Health and Safety Monitoring in Smart Cities" 2024 8th International Conference on I-SMAC, Pages 122-129.

[2] Kalyan Kumar Dasari & Dr, K Venkatesh Sharma, "A Study on Network Security through a Mobile Agent Based Intrusion Detection Framework", JASRAE, vol: 11, Pages: 209-214, 2016.

[3] Dr.K.Sujatha, Dr.Kalyankumar Dasari , S. N. V. J. Devi Kosuru , Nagireddi Surya Kala , Dr. Maithili K , Dr.N.Krishnaveni, " Anomaly Detection In Next-Gen Iot:Giant Trevally Optimized Lightweight Fortified Attentional Convolutional Network," Journal of Theoretical and Applied Information Technology, 15th January 2025. Vol.103. No.1,pages: 22-39.

[4] Kalyankumar Dasari, Dr. K. Venkatesh Sharma, "Analyzing the Role of Mobile Agent in Intrusion Detection System", JASRAE, vol : 15, Pages: 566-573,2018.

[5] Kalyan Kumar Dasari&amp, M Prabhakar, "Professionally Resolve the Password Security knowledge in the Contexts of Technology", IJCCIT, Vol: 3, Issue:1, 2015.

[6] S Deepajothi, Kalyankumar Dasari, N Krishnaveni, R Juliana, Neeraj Shrivastava, Kireet Muppavaram, "Predicting Software Energy Consumption Using Time Series-Based Recurrent Neural Network with Natural Language Processing on Stack Overflow Data", 2024 Asian Conference on Communication and Networks (ASIANComNet), Pages:1-6, Publisher: IEEE.

[7] S Neelima, Kalyankumar Dasari, A Lakshmanarao, Peluru Janardhana Rao, Madhan Kumar Jetty, "An Efficient Deep Learning framework with CNN and RBM for Native Speech to Text Translation", 2024 3rd International Conference for Advancement in Technology (ICONAT), Pages: 1-6,Publisher :IEEE.

[8] A Lakshmanarao, P Bhagya Madhuri, Kalyankumar Dasari, Kakumanu Ashok Babu, Shaik Ruhi Sulthana, "An Efficient Android Malware Detection Model using Convnets and Resnet Models",2024 International Conference on Intelligent Algorithms for Computational Intelligence Systems (IACIS), Pages :1-6, Publisher : IEEE

[9] Dr.D.Kalyankumar, Saranam Kavyasri, Mandadi Mohan Manikanta, Pandrangi Veera Sekhara Rao, GanugapantaVenkata Pavan Reddy, "Build a Tool for Digital Forensics to Analyze and Recover Information from Compromised Systems", IJMTST, Vol: 10, Issue: 02, Pages:173-180, 2024.

[10] Dr.D.Kalyankumar, Kota Nanisai Krishna, Gorantla Nagarjuna, PuvvadaVenkata Naga Sai Jagadesh Kumar, Modepalli Yeswanth Chowdary, "Email Phishing Simulations Serve as a Valuable Tool in Fostering a Culture of Cyber security Awareness", IJMTST, Vol: 10, Issue: 02, Pages:151-157, 2024.

[11] Dr.D.Kalyankumar, Muhammad Shaguftha, Putti Venkata Sujinth, Mudraboyina Naga Praveen Kumar, Namburi Karthikeya, "Implementing a Chatbot with End-To-End Encryption for Secure and Private Conversations", IJMTST, Vol: 10, Issue: 02, Pages:130-136, 2024.

[12] Dr.D.Kalyankumar, Panyam Bhanu Latha, Y. Manikanta Kalyan, Kancheti Deepu Prabhunadh, Siddi Pavan Kumar, "A Proactive Defense Mechanism against Cyber Threats Using Next-Generation Intrusion Detection System", IJMTST, Vol: 10, Issue: 02, Pages:110-116, 2024.

[13] Kalyan Kumar Dasari, K Dr , "Mobile Agent Applications in Intrusion Detection System (IDS)'-JASC, Vol: 4, Issue : 5, Pages: 97-103, 2017.
[14] V.Monica, D. Kalyan Kumar, "BACKGROUND SUBTRACTION BY USING DECOLOR ALGORITHM", IJATCSE, Vol. 3, No.1, Pages: 273 – 277 (2014).
[15] GanugapantaVenkata Pavan Reddy Dr.D.Kalyankumar, Saranam Kavyasri, Mandadi Mohan Manikanta, Pandrangi Veera Sekhara Rao "Build a Tool for Digital Forensics to Analyze and Recover Information from Compromised Systems", IJMTST, Vol: 10, Issue: 02, Pages:173-180, 2024.