Implementation of Secure Shell in Document Sharing Application with TCP/IP-Based Client Server

Della Rizkynta Br Barus[1]
[1]Department of Informatic Engineering, Faculty of Computer and Engineering, University of Harapan Medan, Medan, Indonesia
[1]dellatimika123@gmail.com

Article Info

Article history:

Received August 09, 2025
Revised August 10, 2025
Accepted August 10, 2025

Keywords:

Client server
Transfer
File sharing
Data security
Secure shell

ABSTRACT

The network laboratory module is being reviewed, in which the author conducted data exchange using a LAN cable system and the operating systems provided by Windows or Linux for document transfer and data transmission. The idea is to create software that will perform document transfer and data reception using a sharing medium. The aim is to simplify the process for computer users to send and receive transferred documents. The advantage of client-server software for uploading images and sharing images, documents, and data such as documents and other files is that a flash drive is no longer needed to transfer data, which could potentially contain viruses that could damage the data on the user's computer. Data security is enhanced through secure shell (SSH) connection settings for computers connected via a local area network (LAN). Document sharing security utilises SSH, a cryptographic network protocol designed to securely operate network services over insecure networks that could otherwise endanger data stored on the sender's or recipient's computer.

Corresponding Author:

Della Rizkynta Br Barus
University of Harapan Medan
Email: dellatimika123@gmail.com

1. INTRODUCTION

A client (data recipient) is a computer device that has a system or process that makes a request for data or services to a server (data sender), while a server is a system that can perform processes that provide the data or services requested by the client [1], [2].

This client-server system design is intended for transferring documents, data, images, or software. The author directly observed and reviewed the network laboratory module, where data exchange was conducted using a LAN cable system with operating systems provided by Windows or Linux for sharing (document transfer and data transmission). The idea was to create software that would perform document transfer and data reception via sharing media. The aim is to make it easier for computer users to send and receive transferred documents [3], [4], [5].

Development of client-server software applications for digital transfer from one client to another, to transfer digital data in the form of images, documents, and other data, requires clients that will interact using a LAN (local area network) cable to connect one computer to another, with TCP/IP settings as the computer address identifier [6]. The advantage of client-server software for uploading images, sharing images, documents, and data such as documents and other files is that a flash drive is no longer needed to transfer data [7], [8] which may contain viruses that could damage data on the user's computer [9], [10], [11], [12]. Data security is enhanced through secure shell (SSH) connection settings for computers connected to a LAN [13], [14].

Data security for document sharing uses secure shell technology, which functions as a cryptographic network protocol used to operate network services securely over unsecure networks that could compromise data stored on the sender's or recipient's computers. One of the protocols used in parallel computing is the Network File System (NFS) [15], which is a protocol for sharing resources over a network independently of the type of machine, operating system, or transport protocol used. NFS utilizes Remote Procedure Call (RPC) [16] to allow authorized users to access files on a remote host as if they were on the local system.

According to Fachruddin et al. (2022), in a study titled "Application of Centralized File Sharing Using Samba Server at the Ratu Samban Subdistrict Office," a network system was built using Samba Server as a connector between various operating systems. Samba was installed on the server computer, then a special folder was created to store files that could be accessed by client computers through the download or data processing process. The network used a star topology [17].

Meanwhile, according to Sulistyo & Oktavianto (2020) in their study "Design and Implementation of File Sharing Using Samba Server," the increasing volume of data (big data) requires fast and secure information exchange processes. Samba is one solution because it is open source, free, and provides facilities for sharing data, files, and printers. Although it runs on the Linux/Unix operating system, Samba enables communication between computers with different operating systems, such as Linux and Windows, which use different protocols [18].

Based on the background description above, the author came up with the idea for the research title: "Implementation of Secure Shell in Document Sharing Applications with TCP/IP Based on Client Server."

2. METHOD

2.1. System Analysis

This study discusses the application of a desktop-based system for document sharing with secure shell (SSH) security, as an alternative to the manual method in Windows that requires manual network and IP address settings. This application is designed to simplify and speed up the process of sharing documents between server and client computers. Each computer uses a class C IP address, with SSH-based data security to prevent viruses and file corruption.

2.2. Problem Analysis

In the Windows operating system, documents are transferred between computers via connected IP addresses, but this requires sharing settings that are quite complicated for novice users. As a result, many users choose to transfer documents using flash drives, which are susceptible to viruses and can damage data on hard drives. The sharing process in Windows requires the activation of several settings, making it impractical for new users.

2.3. Problem Solving

The problem solved in this research is to develop client-server-based software to facilitate document transfer between computers, replacing manual methods in Windows or Linux operating systems. Its advantages include eliminating the need for flash drives and improving data security through LAN connections, which have limited range, are easy to develop, and support high transfer speeds.

2.4. Client-Server

Client–Server is a network architecture that separates client applications (GUI-based) from servers, where clients can request data or information from servers. Database systems integrate interrelated data for use by various applications within an organization. Database system types include single user, classic multiuser, and client-server:

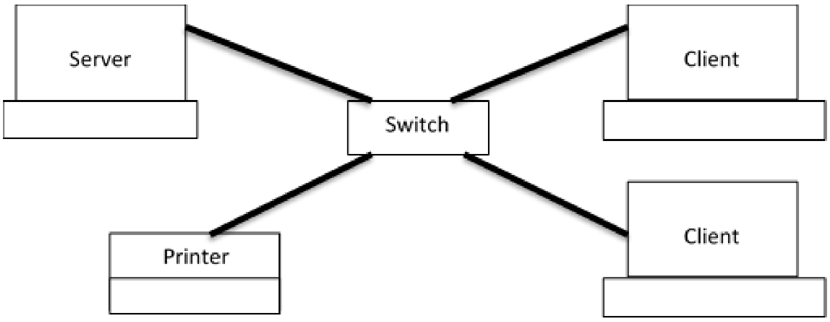


Figure 1. Client Server Topology Display

TCP/IP (Transmission Control Protocol/Internet Protocol) is a set of network protocols that serve as the standard for communication on the internet, enabling computers with different types of hardware and operating systems to exchange data. In a document sharing system, each server and client computer has a specific IP address, which is configured according to IP addressing to support the data transfer process.

Table 1. Computer IP Address

IP Address Class	First octet (decimal)	First octet (binary)	Computer IP Address
Server	1–126	0xxx xxxx	192.168.1.1
Client	128–191	10xx xxxx	192.168.1.2

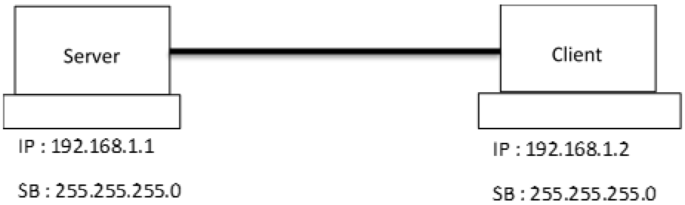


Figure 2. IP Address Addressing Display

2.5. System Design

This program is designed to manipulate digital images according to user needs, allowing users to select images and apply various manipulation options through the available menu.

Use cass diagrams

Use case diagrams illustrate the interactions between the system and actors, describing the types of interactions and how the system appears from the user's perspective.

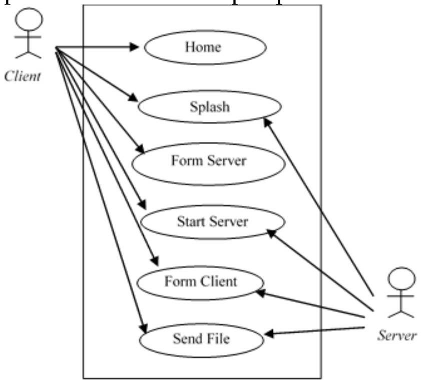


Figure 3. System Use Case Design

Activity diagrams are used to describe the flow of system behavior, similar to the use of flowcharts.

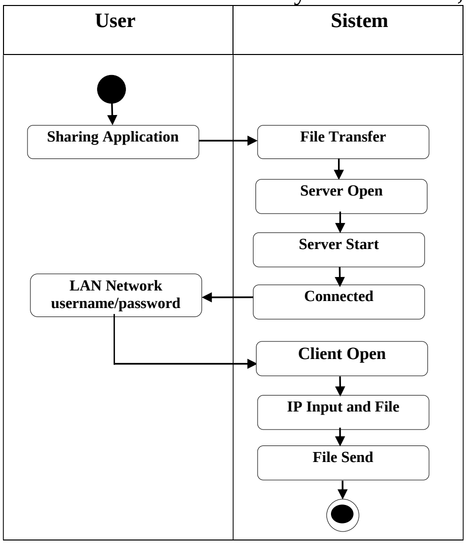


Figure 4. Activity Diagram of the System

2.6. System Flowchart

The system design uses a flowchart system as shown in the image below:

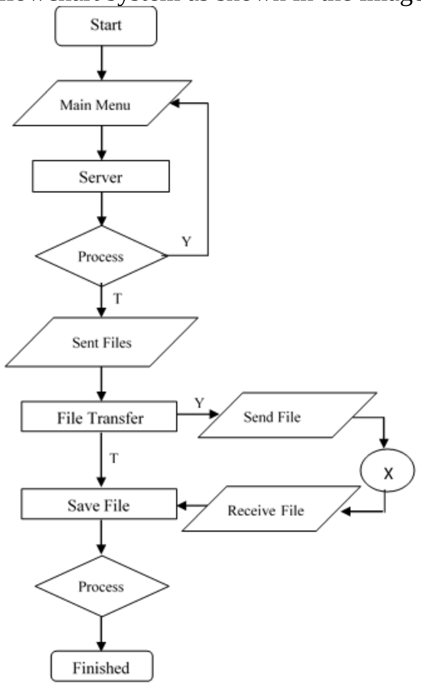
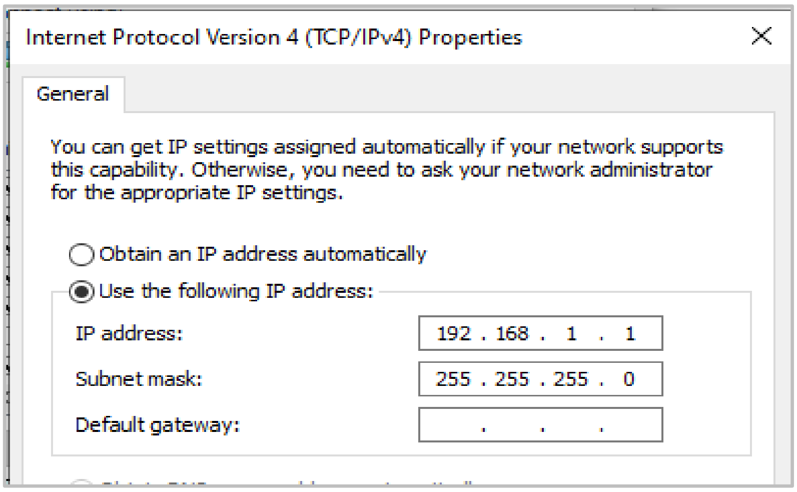


Figure 5. System Flowchart Display

3. RESULTS AND DISCUSSION

3.1. Results

The file transfer application operates on a peer-to-peer (P2P) network architecture, where each client simultaneously functions as both the sender and the server. To facilitate the data transfer process, computers are interconnected via a Local Area Network (LAN), utilizing an RG45 network cable as the physical connection medium. The system's performance was evaluated through testing on a computer with the following hardware specifications: an Intel Core i3 processor at 2.27 GHz, a 320 GB hard drive, 2.00 GB of memory, an onboard VGA card, and a 14-inch LCD monitor, along with a standard optical mouse and keyboard. The tests were conducted within a software environment consisting of the Microsoft Windows 7 operating system, PHP, and MySQL. System testing was performed to ensure the application aligns with its original objectives and to assess its overall performance, the results of which are detailed in the subsequent figures.


Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

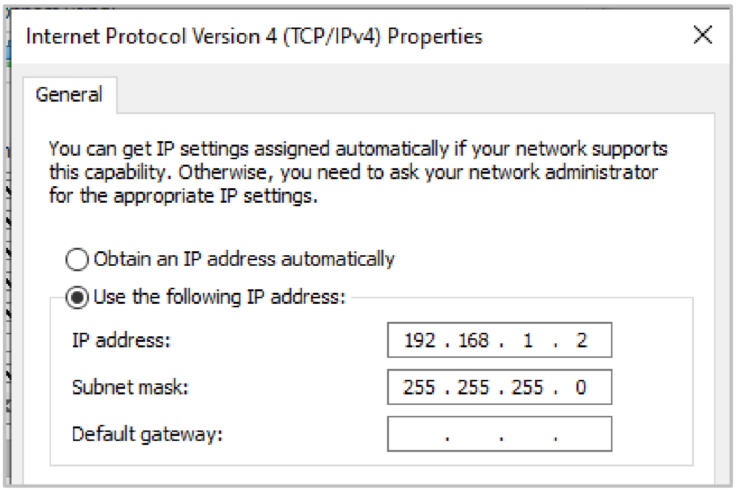
IP address: 192 . 168 . 1 . 1

Subnet mask: 255 . 255 . 255 . 0

Default gateway: . . .

Figure 6. IP Address Server

Figure 6 displays the static IP address configuration for the computer designated as the Server. The server is assigned the IP address 192.168.1.1. This fixed address serves as the connection endpoint for clients wishing to access the document sharing service.



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address:	192 . 168 . 1 . 2
Subnet mask:	255 . 255 . 255 . 0
Default gateway:	. . .

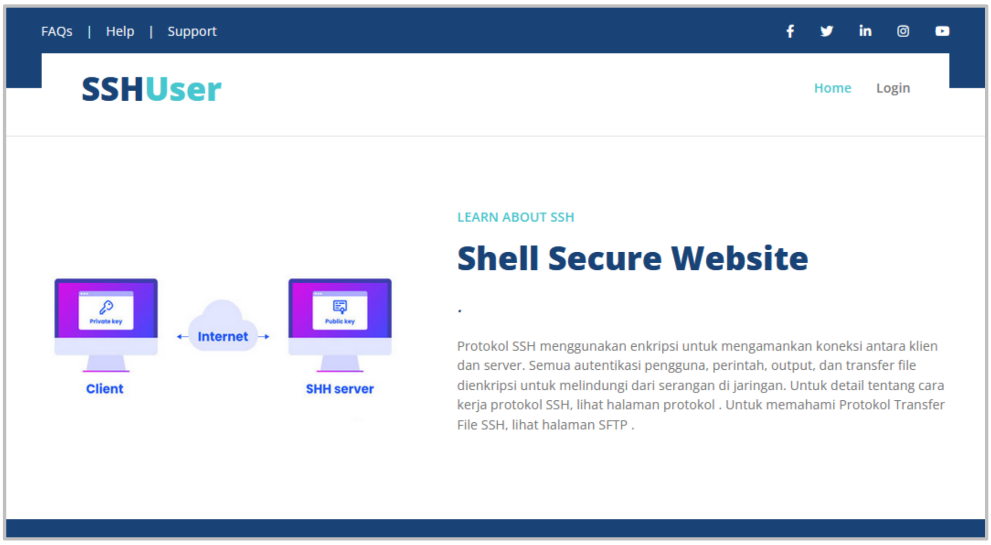
Figure 7. IP Address Client

Figure 7 shows the IP address configuration for the Client computer, which is set to 192.168.1.2. Both computers use the same Subnet Mask (255.255.255.0), indicating that they operate on the same local network. This setup is essential as it enables the Client (192.168.1.2) to connect directly to the Server (192.168.1.1) via the TCP/IP protocol. This direct connection is what will subsequently be secured using Secure Shell (SSH) encryption during the document sharing process.

3.2. Discussion

Implementation is the result of a design that becomes an application program that can be operated and achieves results in accordance with the design. After conducting the analysis and design stages, the next step is to achieve the results of the software that has been created.

The file transfer application system uses a peer-to-peer computer network system, with the client acting as the sender and also functioning as the application server. The connection links one computer to another to enable data transfer between connected computers using a LAN (local area network) with an RG 45 network cable as the connection device.



LEARN ABOUT SSH

Shell Secure Website

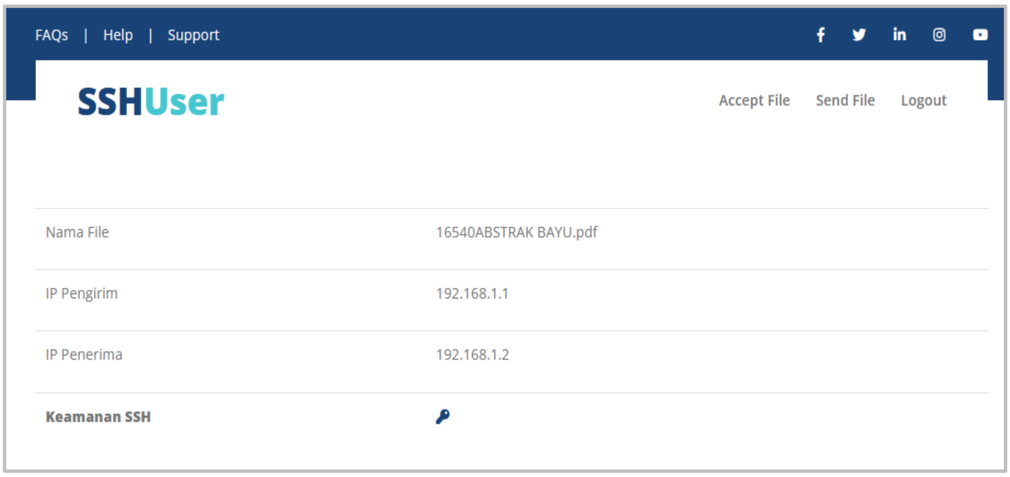
.

Protokol SSH menggunakan enkripsi untuk mengamankan koneksi antara klien dan server. Semua autentikasi pengguna, perintah, output, dan transfer file dienkripsi untuk melindungi dari serangan di jaringan. Untuk detail tentang cara kerja protokol SSH, lihat halaman protokol . Untuk memahami Protokol Transfer File SSH, lihat halaman SFTP .

Figure 8. Home User

The image caption explains that Secure Shell (SSH) is used as a security protocol in the file transfer process. Every user who accesses the system is registered and has authentication information that includes an

IP address, username, and special password. Additionally, the description includes information about the files intended for each user, ensuring that only verified users can receive or access the specified files. The use of SSH in this mechanism serves to encrypt data during transmission, maintain information confidentiality, and prevent unauthorized access from external parties. The file acceptance form is designed to display the file sending process form page in the application, as shown in the image below:



Nama File	16540ABSTRAK BAYU.pdf
IP Pengirim	192.168.1.1
IP Penerima	192.168.1.2
Keamanan SSH	🔑

Figure 9. View Form Accept File

In the accept file display design, it is shown that every file or document sent by the server can be viewed by users through the available interface. However, users cannot directly open or download the file. To access them, users must first enter the security password generated and sent by the server via the Secure Shell (SSH) method. Only after this authentication process is successful will the system grant the user permission to download the specified document. This mechanism is designed to ensure that every file transferred can only be accessed by authorized parties, thereby maintaining data security throughout the distribution process.

The implementation of Secure Shell (SSH) in a client-server architecture-based document sharing application using the TCP/IP protocol demonstrates that the integration of this security protocol provides significant protection for transferred data. Based on the tests conducted, the use of SSH effectively encrypts data during the transmission process, ensuring that information sent over the network cannot be easily accessed or modified by unauthorized parties. This aligns with previous research findings stating that SSH is one of the most reliable protocols in maintaining the confidentiality and integrity of data in network communications.

On the server side, the system is designed to manage user authentication through IP address registration, username, and password, which are directly linked to the SSH security mechanism. Every document access request from the client is verified first before being allowed to proceed with the download process. This process ensures that only users with valid credentials can access specific documents. This approach is effective in preventing unauthorized access and man-in-the-middle attacks.

Additionally, testing in a local area network (LAN) environment shows that SSH implementation does not significantly impact document transfer speeds. Files in various formats, such as Microsoft Word, Excel, PDF, and image files, can be transferred with relatively short response times despite the encryption process. This demonstrates that the use of security protocols does not necessarily negatively impact system efficiency, provided that network and software optimization is properly implemented. From a system design perspective, the use of PHP and HTML for the user interface, along with MySQL as the database, provides flexibility in managing user files and data. The client-server architecture enables clear workload distribution, where the server acts as the central hub for file management and authentication, while the client serves as the recipient interacting directly with the application interface.

However, this research still has room for further development. For example, the system can be expanded to support wireless networks while considering more complex security challenges. Additionally, logging features can be added to record all file transfer activities as a preventive measure against future security incidents. Thus, the results of this implementation demonstrate that integrating SSH into a TCP/IP-based document-sharing application and client-server architecture can significantly enhance data security without compromising system performance, and has the potential to be adopted in organizational or institutional environments requiring secure and efficient file transfer mechanisms.

4. CONCLUSION

Based on the results of research and testing that has been conducted, it can be concluded that the client-server architecture-based file transfer software with LAN cable connection has been successfully developed and functions according to its design objectives. This system enables the secure transmission of shared files (file sharing) using the secure shell (SSH) method as a security protocol, thereby maintaining data integrity and confidentiality during the transmission process.

The application demonstrates the ability to operate optimally on a local area network (LAN), supporting various document formats such as Microsoft Word (.docx), Microsoft Excel (.xlsx), Portable Document Format (.pdf), and image files in common formats (.jpg, .png). The system design and implementation were carried out using the PHP programming language and HTML as the user interface, along with MySQL as the database management system to organize and store information related to the file transfer process.

Overall, the development of this software not only meets the need for fast and secure file transfer in a local network environment but also provides a foundation that can be further developed, such as by adding wireless network support, optimizing transfer speed, and integrating stronger encryption features to enhance security levels. As such, the results of this research have the potential to make a significant contribution to the efficiency and security of data exchange in organizational or educational institutional environments.

REFERENCES

[1] G. Nyabuto, “Client-server Architecture, a Review,” *International Journal of Advanced Science and Computer Applications*, vol. 3, no. 2, Jan. 2024, doi: 10.47679/IJASCA.V3I1.48.

[2] Y. Klushyn, “Specialized Software Platform for Analysis of Information in Data Stores,” *Computer systems and network*, vol. 6, no. 2, pp. 93–106, Dec. 2024, doi: 10.23939/CSN2024.02.093.

[3] S. M. Prasetiyo and Y. Asri, “Implementation of File Sharing and Remote Desktop Connection (RDC) by Utilizing Local Area Network (LAN),” *Jurnal Inotera*, vol. 9, no. 2, pp. 314–322, Aug. 2024, doi: 10.31572/INOTERA.VOL9.ISS2.2024.ID366.

[4] D. A. J. Al-Khaffaf and M. G. Al-Hamiri, “Performance evaluation of campus network involving VLAN and broadband multimedia wireless networks using OPNET modeler,” *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 19, no. 5, pp. 1490–1497, Oct. 2021, doi: 10.12928/TELKOMNIKA.V19I5.18531.

[5] K. J, N. J.N, S. K.O, I. M. E, and I. N. E, “Optimizing Local Area Network Performance: Insight from Riverbed Modelling,” *Journal of Engineering Research and Reports*, vol. 26, no. 6, pp. 212–222, May 2024, doi: 10.9734/JERR/2024/V26I61175.

[6] R. K. Rithvik and A. Mohan, “Fingerprint Password Method Provides Improved Accuracy over Token-based Authentication for Efficient and Secure File Transfers,” *E3S Web of Conferences*, vol. 491, p. 04023, Feb. 2024, doi: 10.1051/E3SCONF/202449104023.

[7] admin admin and R. Al King, “Data Security in Cloud Computing,” *International Journal of Wireless and Ad Hoc Communication*, vol. 7, no. 1, pp. 50–61, 2023, doi: 10.54216/IJWAC.070105.

[8] P. Raja, S. Reddy, and K. Ravindranath, “Enhancing Secure and Reliable Data Transfer through Robust Integrity,” *Journal of Electrical Systems*, vol. 20, no. 1s, pp. 900–910, Mar. 2024, doi: 10.52783/JES.841.

[9] E. Irmak, E. Kabalci, and Y. Kabalci, “Digital Transformation of Microgrids: A Review of Design, Operation, Optimization, and Cybersecurity,” *Energies 2023, Vol. 16, Page 4590*, vol. 16, no. 12, p. 4590, Jun. 2023, doi: 10.3390/EN16124590.

[10] P. M. Datta and T. Acton, “Did a USB drive disrupt a nuclear program? A Defense in Depth (DiD) teaching case,” *Journal of Information Technology Teaching Cases*, vol. 14, no. 2, pp. 311–321, Nov. 2024, doi: 10.1177/20438869231200284;PAGE:STRING:ARTICLE/CHAPTER.

[11] P. Weichbroth, K. Wereszko, H. Anacka, and J. Kowal, “Security of Cryptocurrencies: A View on the State-of-the-Art Research and Current Developments,” *Sensors 2023, Vol. 23, Page 3155*, vol. 23, no. 6, p. 3155, Mar. 2023, doi: 10.3390/S23063155.

[12] R. Spolaor, H. Liu, F. Turrin, M. Conti, and X. Cheng, “Plug and Power: Fingerprinting USB Powered Peripherals via Power Side-channel,” *Proceedings - IEEE INFOCOM*, vol. 2023-May, 2023, doi: 10.1109/INFOCOM53939.2023.10229048.

[13] Hadya. S. Hawedi, O. A. Bentaher, and K. E. I. Abodhir, “REMOTE ACCESS TO A ROUTER SECURELY USING SSH,” *Journal of the Academic Forum*, vol. 5, no. 1, pp. 174–189, Jan. 2021, doi: 10.59743/JAF.V5I1.177.

[14] H. Touil, F. Hdioud, N. E. L. Akkad, and K. Satori, “The Security of SSH Protocol Public Key Sharing in the Post-Quantum Era,” *International Journal of Computing*, vol. 23, no. 3, pp. 317–323, Oct. 2024, doi: 10.47839/IJC.23.3.3650.

[15] F. Boito, G. Pallez, and L. Teylo, “The role of storage target allocation in applications’ I/O performance with BeeGFS,” *Proceedings - IEEE International Conference on Cluster Computing, ICCC*, vol. 2022-September, pp. 267–277, 2022, doi: 10.1109/CLUSTER51413.2022.00039.

[16] A. Pourhabibi, M. Sutherland, A. Daglis, and B. Falsafi, “Cerebros: Evading the RPC tax in datacenters,” *Proceedings of the Annual International Symposium on Microarchitecture, MICRO*, pp. 407–420, Oct. 2021, doi: 10.1145/3466752.3480055;CSUBTYPE:STRING:CONFERENCE.

[17] P. File Sharing Terpusat Menggunakan Samba Server Pada Kantor Kecamatan Ratu Samban, R. Fachruddin, and E. Prasetiyo Rohmawan, “Penerapan File Sharing Terpusat Menggunakan Samba Server Pada Kantor Kecamatan Ratu Samban,” *JURNAL MEDIA INFOTAMA*, vol. 18, no. 2, pp. 197–207, Oct. 2022, doi: 10.37676/JMI.V18I2.2657.

[18] H. W. Sulistyo and H. Oktavianto, “Perancangan dan Implementasi File Sharing menggunakan Samba Server,” *Jurnal Aplikasi Sistem Informasi dan Elektronika*, vol. 2, no. 1, pp. 24–30, Jul. 2020, doi: 10.32528/JASIE.V2I1.4039.