Journal of Ubiquitous Computing and Communication Technologies (ISSN: 2582-337X)
www.irojournals.com/jucct/

Authentication in Peer-to-Peer Cloud using AES and SRP

Sharon Rose H[1], Razeenath Aafiya A[2], Kamachi Swetha R[3], Ramya G[4]

[123] Student, Dept of Information Technology, National Engineering College, Kovilpatti, India

[4] Asst.Professor, Dept of Information Technology, National Engineering College, Kovilpatti, India

E-mail: [1]2015019@nec.edu.in, [2]2015014@nec.edu.in, [3]2015038@nec.edu.in, [4]ramya-it@nec.edu.in

Abstract

The system focuses on proposing a robust authentication framework based on these Secure Remote Password (SRP) protocol to enhance the security of data exchange in Peer-to-Peer (P2P) cloud environments. The Advanced Encryption Standard (AES) algorithm provides encryption and the SRP protocol leverages cryptographic primitives to establish mutual authentication between cloud servers and users, ensures that data, remains protected against any vulnerabilities. The system encompasses various stages, including user registration, key establishment, and secure data transmission. After user registration, the SRP protocol employs a zero-knowledge proof mechanism to maintain the security of data available on cloud servers, mitigating the risks associated with password leaks and unauthorized access. Furthermore, the protocol facilitates secure key exchange to establish a confidential communication channel between peers, enabling encrypted data transmission. The system integrates SRP protocol and AES encryption, fortifying data security in P2P cloud environments through robust authentication, ensuring a comprehensive defence against potential vulnerabilities.

Keywords: SRP, P2P, AES, Zero Knowledge, Encryption.

1. Introduction

Cloud computing has revolutionized the way data is stored, processed, and accessed, providing users with a scalable and cost-effective solution for managing digital assets. However, traditional centralized cloud architectures raise concerns regarding data privacy, security, and potential single points of failure. To address these security concerns, researchers

DOI: https://doi.org/10.36548/jucct.2023.4.003
Received: 09.10.2023, received in revised form: 02.11.2023, accepted: 16.11.2023, published: 28.11.2023

had given innovative solutions, one of which is the P2P Cloud Authentication and Key Agreement Scheme (KAS)using cryptographic techniques. The scheme aims to establish secure and efficient authentication mechanisms and key agreement protocols tailored for P2P cloud computing environments. The primary goal of the P2P cloud authentication and key agreement scheme is to enable secure communication between peers while ensuring data confidentiality and integrity. The scheme leverages cryptographic techniques to authenticate users and establish shared secret keys that enable encrypted communication channels among peers. By utilizing the approach, the P2P cloud authentication and key agreement scheme using cryptographic techniques establishes a robust and decentralized security infrastructure, mitigating the risks associated with central points of failure and unauthorized access.

1.1 Peer To Peer Cloud

Peer To Peer cloud architecture represents a decentralized network where participating clouds collaborate to store and manage data and services. Unlike traditional client-server models, P2P clouds do not rely on a central authority for authentication and data storage. Instead, data is distributed across multiple clouds, making it more resilient to failures and reducing the risk of data loss. Each cloud in the P2P cloud acts both as a provider and consumer of resources, contributing to the overall stability and scalability of the system. The decentralized nature of P2P cloud ensures high fault tolerance. If one cloud fails, the network redistributes the data and services to other available clouds, reducing the impact of cloud failures on the overall system. clouds are highly scalable. As more clouds join the network, the system's capacity and resources increase accordingly, allowing the cloud to grow without reliance on a single data center. Data is distributed and replicated across multiple clouds, ensuring redundancy and data availability even in the face of cloud failures or network disruptions. It employs load balancing techniques to distribute the workload evenly across participating clouds, optimizing resource utilization and preventing overburdening of specific clouds. The users can interact with the network without revealing their identity, providing a level of anonymity and privacy.

1.2 Advanced Encryption Standard (AES)

Advanced Encryption Standard (AES) stands as one of the most widely used and secure cryptographic algorithms in the world. It was established as a standard by the National Institute of Standards and Technology (NIST) in 2001, supplanting the Data Encryption Standard (DES)

which had become vulnerable to increasingly powerful computing capabilities. AES is a symmetric-key algorithm, meaning it uses the same key for both encryption and decryption processes. The key is a binary string of a specific length, which can be either 128, 192, or 256 bits. AES operates on blocks of data, dividing them into fixed-size blocks and then applying a series of transformations. The strength of AES lies in its ability to obfuscate the relationship between the input and output blocks, rendering it highly resistant to cryptographic attacks. It employs a substitution-permutation network, combining processes of substitution (replacing elements) and permutation (rearranging elements) in multiple rounds. These rounds depend on the key length: 10 rounds for a 128-bit key, 12 rounds for a 192-bit key, and 14 rounds for a 256-bit key. One of the fundamental reasons for AES's effectiveness is the complexity of the operations involved. The algorithm employs a substitution box (S-box) to substitute bytes, creating non-linear relationships within the data. Additionally, AES utilizes a key expansion process, where the original key is transformed into a series of round keys through a sequence of operations. This makes it exceedingly difficult for adversaries to reverse engineer the original key from the round keys. 19 AES is widely adopted across various domains, from securing sensitive data in financial transactions, safeguarding communications over the internet, to protecting classified government information. Its versatility and robustness have made it a cornerstone of modern cryptography. Despite its widespread use, AES has withstood extensive scrutiny from the global cryptographic community, solidifying its reputation as a highly secure encryption standard

1.3 Secure Remote Password (SRP) Protocol

The Secure Remote Password (SRP) protocol is a strong cryptographic authentication mechanism that operates based on a combination of cryptographic hashing and public-key cryptography. Unlike traditional password-based authentication, SRP does not send the user's password over the network during the authentication process, thereby mitigating potential eavesdropping and replay attacks. Instead, SRP uses a series of mathematical operations and shared secret values to establish a secure and authenticated session between the user and the server (or cloud in the case of P2P cloud). SRP follows a challenge-response authentication scheme. When a user initiates the login process, the cloud challenges the user to prove their identity without revealing their actual password. This is achieved through the exchange of randomly generated values and cryptographic calculations. The result is a mutual authentication process where both the user and the cloud verify each other's identity, ensuring a secure communication channel. The aspect of the SRP protocol is that compromised

verification keys are of little value to an attacker. Possession of a verification key does not allow a user to be impersonated and it cannot be used to obtain the user's password except by way of a computationally infeasible dictionary attack. A compromised key would, however,allow an attacker to impersonate the server side of an SRP authenticated connection. Consequently, care should be taken to prevent unauthorized access to verification keys for applications in which the client-side relies on the server being genuine.

2. Related Work

Hong Zhong et al., [1] The paper underscores the pivotal role of robust authentication mechanisms in fostering trust during data migration processes within cloud computing. Data migration, involving the transfer of data between systems or cloud providers, is a fundamental aspect of optimizing resources and enhancing scalability. Robust authentication mechanisms become crucial components in this trust-building process.

Nesrine Kaaniche et al., [2] The paper appears to focus on security aspects related to cloud data storage. Specifically, it emphasizes the use of cryptographic mechanisms to enhance the security of data stored in the cloud. It delves into various cryptographic techniques employed for securing data in cloud storage. This could include encryption algorithms, key management, access control, and other cryptographic protocols.

Sarojini G et al., [3] The Enhanced Mutual Trusted Access Control Algorithm (EMTACA) places a strong emphasis on mutual trust between cloud users and service providers. EMTACA employs reputation aggregation and integration to enhance authentication processes, emphasizing the critical role of trust in accessing cloud services

Mazhar Ali et al., [4] Ensuring data confidentiality and secure cross-cloud data migration are central concerns in cloud security. Secure Data Sharing in Clouds (SeDaSC) methodology enables secure data sharing within a group using cloud storage. This methodology not only addresses insider threats and access control but also maintains forward and backward security. Moreover, it is designed to reduce the computational burden on mobile devices, making it compatible with resource-constrained environment

Jia et al., [5] Secure authentication within resource-constrained Mobile Edge Computing (MEC) environment introduces an identity-based anonymous authentication scheme that prioritizes both security and privacy in authentication processes, contributing a

valuable solution for securing access in such settings.

Cui et al.,[6] An authentication scheme for vehicular networks in multi-cloud environments. This survey underscores the importance of encryption techniques and access control in maintaining data confidentiality and security. Together, these surveys emphasize the critical role of encryption methods in ensuring data confidentiality within cloud environments and the significance of authentication in establishing trust and security during cross-cloud data migration

Gupta et al., [7] The research focuses on addressing critical aspects of data security within cloud computing. With a dual emphasis on secure data storage and sharing, the study delves into the unique challenges posed by cloud environments in safeguarding data throughout its lifecycle. Strategies for secure data storage are explored, encompassing encryption methodologies, access controls, and redundancy measures to protect data at rest.

Zhang et al.,[8] The concept of Attribute-Based Keyword Search (ABKS) with cryptographic reverse firewalls (CRF), demonstrates its effectiveness in enabling secure searches over encrypted cloud storage. This approach leverages advanced cryptographic methods, including verifiable functional encryption and non-interactive zero-knowledge proofs, to ensure robust data protection

Shabbir et al.,[9] The Modular Encryption Standard (MES) is showcased as a vital tool for securing sensitive health information and its adoption is proven to yield superior performance compared to other encryption algorithms, reinforcing its role in enhancing data security within healthcare settings

Nhlabatsi et al.,[10] The paper suggests a targeted examination of security risks in cloud computing with a focus on threat-specific assessments. This likely involves a comprehensive analysis of various threats that could potentially compromise the security of cloud environments. The paper might delve into identifying and evaluating specific threats such as data breaches, unauthorized access, and other cyber threats

3. Proposed Work

Authentication is a critical aspect of any cloud-based system, ensuring that only authorized users can access sensitive data and services. The traditional client-server model for authentication has its limitations, such as centralization, potential single points of failure, and scalability issues. To overcome these challenges, the use of P2P cloud architecture combined

with the SRP protocol offers a promising approach. It delves into the concept of authentication using a P2P cloud with SRP. SRP protocol offers a promising approach. It delves into the concept of authentication using a P2P cloud with SRP. The Figure 3.1 shows the secure message flow (SMF) between P2P clouds and the user.

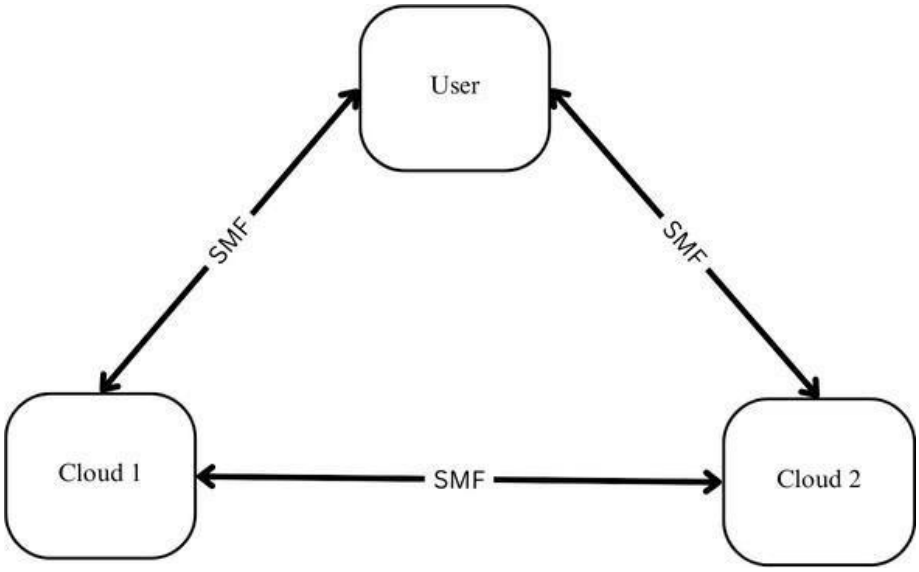

Figure 3.1 BLOCK Diagram of the Authentication using P2P Cloud using SRP

3.1 Public-Key Cryptography for Session Establishment:

1. **Key Exchange:**

   A common method for key exchange is the Diffie-Hellman key exchange, which allows two parties to agree on a shared secret over an untrusted network.

2. **Public and Private Keys:**

   Each user and the cloud entity will have a pair of public and private keys.

3. **Diffie-Hellman Key Exchange:**

   Choose a large prime ppp and a primitive root modulo ppp, denoted as ggg. Each side generates a private key (aaa for the user, bbb for the cloud) and computes a public key ($A = g^a \bmod p$ for the user, $B = g^b \bmod p$ for the cloud).The shared secret (SSS) is computed as $S = B^a \bmod p = A^b \bmod p$.

4. **AES Algorithm (Public-Key Cryptography):**

   Key generation: Select two large prime numbers ppp and qqq, compute n=pqn = pqn=pq, and select eee such that 1<e<ϕ(n)1 < e < \phi(n)1<e<ϕ(n) and eee is coprime to ϕ(n)\phi(n)ϕ(n).

   The public key is (n,e)(n, e)(n,e) and the private key is (n,d)(n, d)(n,d), where ddd is the modular multiplicative inverse of eee modulo ϕ(n)\phi(n)ϕ(n).

5. **Secure Session:**

   The agreed-upon secret is used to derive session keys for symmetric encryption, ensuring that the subsequent communication is secure**.**

3.2 Secure Remote Password (SRP) Protocol

The Secure Remote Password (SRP) protocol is a strong cryptographic authentication mechanism that operates based on a combination of cryptographic hashing and public-key cryptography.

1. **User Registration:**

   The user registers with the server by creating a verifier based on their password using a one-way hash function.

2. **Authentication:**

   During authentication, the user and server engage in a series of computations without transmitting sensitive information.

3. **Exponentiation:**

   Modular exponentiation is commonly used. For example, calculating v=gxmod Nv = g^x \mod Nv=gxmodN, where ggg is a generator, xxx is a private key, and NNN is a large prime.

4. **Hash Functions:**

   Hash functions are used to derive keys from passwords securely.

3.3 Cryptographic Hashing:

1. **Password Hashing:**

   During user registration in SRP, a cryptographic hash function (e.g., SHA-256) is applied to the password to generate a verifier.

2. **Verifying Identity:**

   When authenticating, the user proves their identity by providing responses to challenges based on the verifier, without revealing the password.

3.4 Authentication Process

Once the user is registered, the initiation of the authentication process occurs whenever they wish to access the P2P cloud. The authentication process uses the SRP protocol to establish a secure session between the user and the cloud.

1. **Login Request:**

The user sends a login request to the cloud, indicating their desire to access the P2P cloud. Let's represent this as a function where U is the set of all users, and u is the specific user's username.

LoginRequest(u)

This function signifies the user's intention to log in and includes their username as a parameter.

2. **Challenge Generation:**

Upon receiving the login request, the cloud generates a random challenge, the "verifier." Let's represent this process as a function Generate Verifier(), which generates a random number or mathematical expression.

Verifier = GenerateVerifier()

The generated verifier serves as a cryptographic puzzle for the user.

3. **Session Key Exchange**

The user sends the computed session key back to the cloud as a response to the challenge. The session key is essential for establishing a secure communication channel between the user and the cloud.

4. **Session Key Calculation**

Using the verifier, the username, and the password, the user performs a series of cryptographic calculations to derive a client session key. The calculation involves modular exponentiation and other cryptographic operations, ensuring that the session key is secure and resistant to attacks.

1. **Initialization:**

Let V be the verifier received from the cloud.

Let U be the username of the user.

Let P be the user's password.

2. **Concatenation and Hashing:**

Concatenate the verifier, username, and password: $S = V+U+P$

Hash the concatenated value: $H = \text{Hash}(S)$

3. **Modular Exponentiation:**

Use modular exponentiation to derive a session key. Let g be a generator, and p be a large prime number.

$$K=g^{H} \bmod p$$

The result K is the session key.

The complete mathematical representation can be expressed as:

$$K= g^{\text{Hash}(V+U+P)} \bmod p$$

This process ensures that the session key is derived from a combination of the verifier, username, and password, and the use of modular exponentiation enhances the security of the key. The hashing step helps in creating a fixed-size input for the modular

exponentiation operation. The values of g and p would be predefined parameters in the cryptographic system.

3.5 Verification

Upon receiving the session key, the cloud performs the same cryptographic calculations as the user did during the session key calculation step. If the session key provided by the user matches the one calculated by the cloud, the user's identity is verified, and the cloud grants access to the P2P cloud.

1. **Cloud Verification:**

   Let K_{user} be the session key provided by the user.

   The cloud performs the same cryptographic calculations on its end using the verifier, username, and password, to calculate K_{cloud}.

2. **Verification Equation:**

   The cloud verifies if the session key provided by the use (K_{user}) matches the one calculated by the cloud (K_{cloud}).

   Verification: $K_{user} = K_{cloud}$

3. **Cryptographic Calculations:**

   The cloud repeats the same series of cryptographic calculations performed by the user during the session key calculation step.

   $K_{cloud} = g^{Hash(V+U+P)} \bmod p$

   where g, V,U, P, p, and Hash have the same meanings as in the session key calculation step.

4. **Comparison:**

   If K_{user} and K_{cloud} match, the user's identity is verified, and access is granted.

   The overall verification process can be expressed mathematically as:

   Verification: $K_{user} = g^{Hash(V+U+P)} \bmod p$.

4. Results and Discussion

P2P cloud refers to a user directly sharing computing resources and data with each other in a decentralized manner. Client and server virtual machines (VMs) are established for a secure channel for exchanging data. It handles authentication request from an external source and interface with the internal authentication infrastructure. By using Google Cloud Platform (GCP), Cloud Storage bucket for each virtual machine (VM) serves as an intermediary for seamless data transfer between different cloud environments. Each VM can independently access and manipulate the data within its assigned bucket, allowing for parallel and distributed processing. This approach proves valuable for various purposes such as data staging, data processing, and data backup. For data encryption, the AES algorithm is employed. AES ensures that all data transmitted and stored in the cloud remains encrypted. The generation of encryption keys demands the use of a secure random number generator. It ensures the secure storage of encrypted files within cloud storage or the chosen storage solution. This practice guarantees unpredictability and resilience against cryptographic attacks, reinforcing the security of AES encryption. To authenticate file transfers within the P2P cloud service, the SRP protocol is implemented, guaranteeing that only authorized users can initiate and complete data exchanges, thwarting unauthorized access. Python, a versatile programming language, becomes a tool of choice for the implementation of encryption and authentication processes. Diverse data types, such as .png, .pdf, .mp3, .jpeg, .csv, and more, can be seamlessly shared within this P2P cloud environment. Regardless of the file format, the robust encryption and authentication mechanisms persistently safeguard the user's data against potential security threats.

The Figure 4.1 and Figure 4.2 explains the creation of cloud storage bucket for each VM as an intermediary of transferring data from one cloud to another cloud. This is useful for data staging, data processing and data backup.

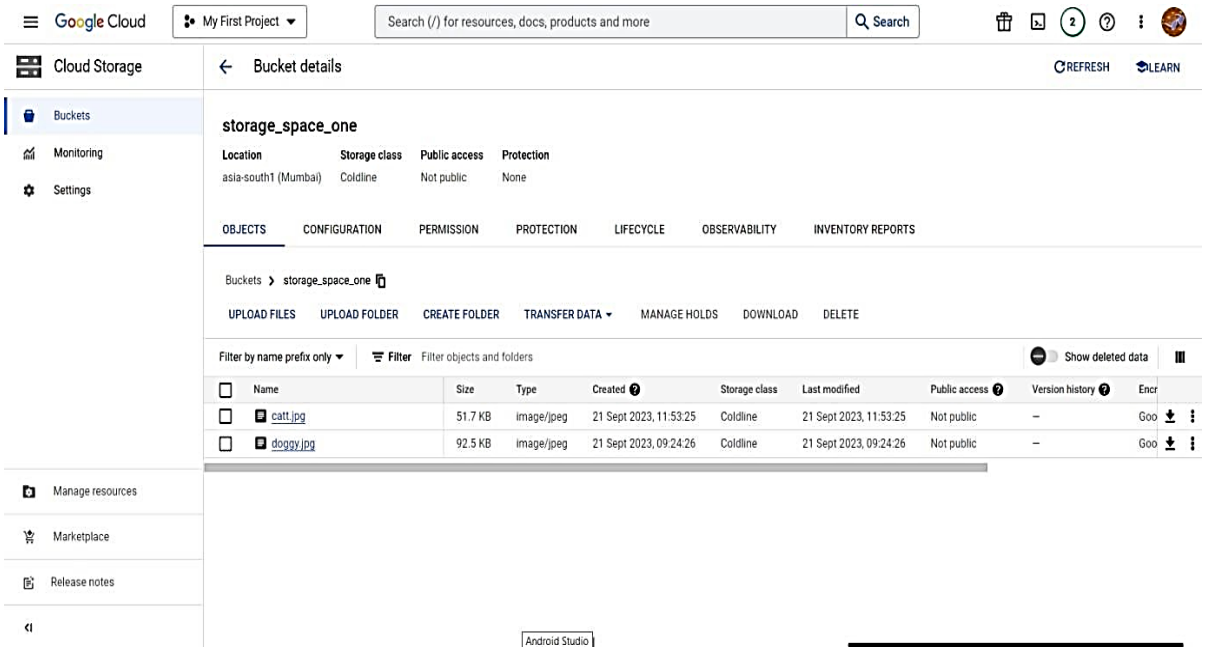

Figure 4.1 Storage Space for Cloud 1

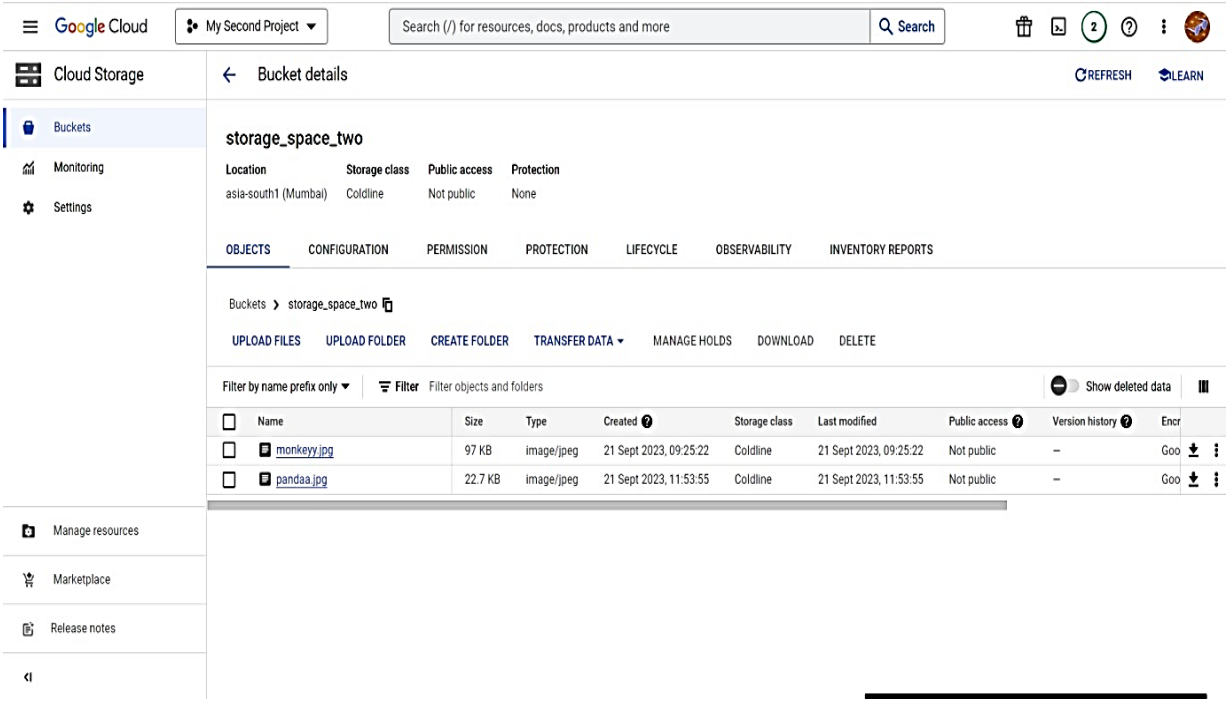

Figure 4.2 Storage Space for Cloud 2

The Figure 4.3 explains the client communication with the server VM by using SSH (secure shell) to connect the server VM instance. It allows securely to access and manage VM remotely.

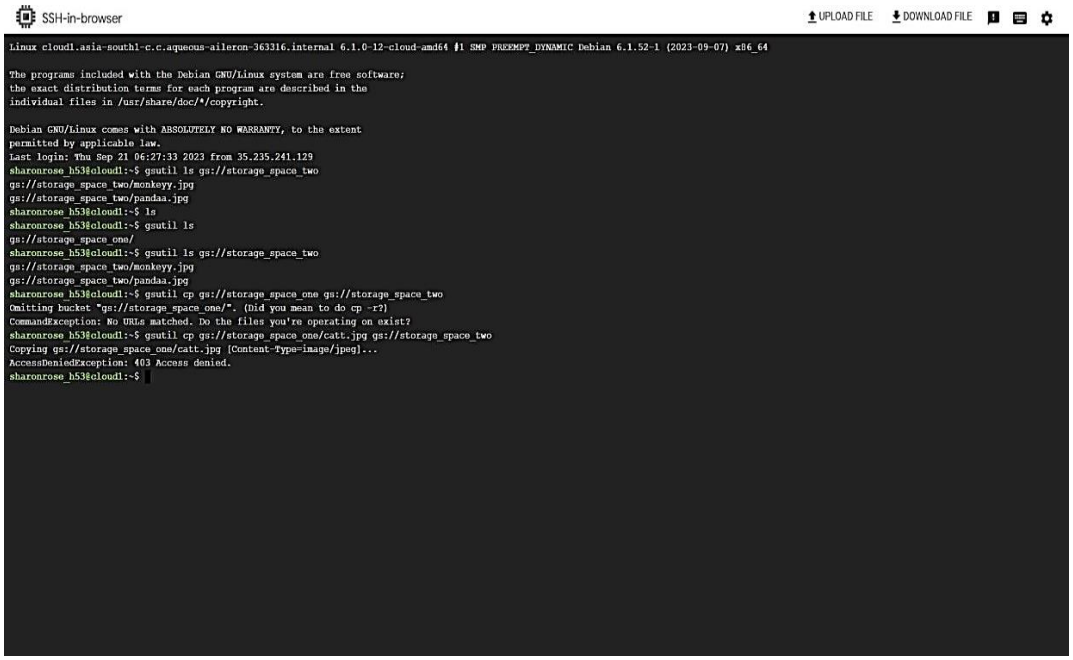

Figure 4.3 Client Communication using SSH

The Figure 4.4 shows the cloud bucket after the data transfer is completed.

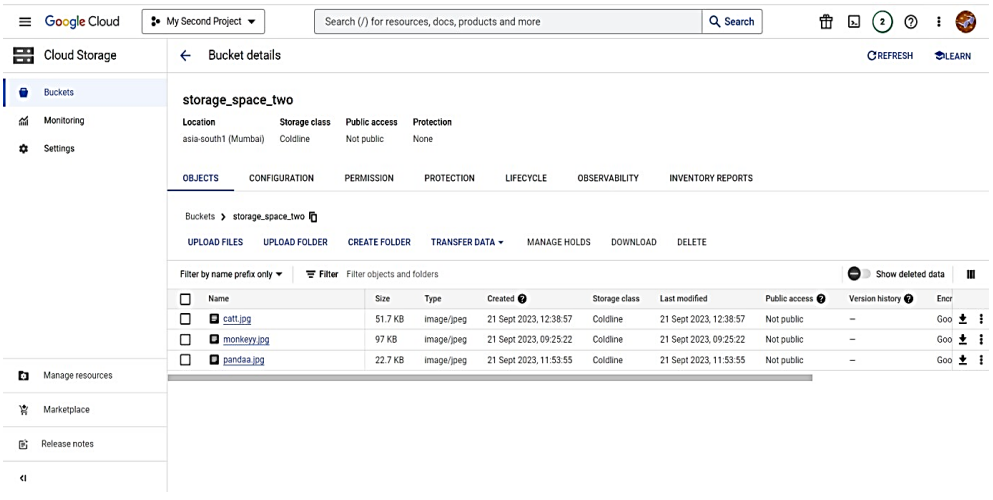

Figure 4.4 File Transferred from Cloud 1 to Cloud 2

4.1 Advantages:

In a P2P cloud, the authentication process is distributed across multiple nodes, reducing the risk of a central server becoming a bottleneck or target for malicious attacks. Each peer in the network can contribute to the authentication process, enhancing scalability and resilience. SRP eliminates the need to transmit passwords over the network, even in hashed form. Instead, it uses a zero-knowledge proof, allowing the user and server to prove knowledge of the password without revealing it. This significantly reduces the risk of password interception or attacks based on intercepted credentials. The protocol involves a series of cryptographic challenges and responses between the user and the server. This exchange is resistant to eavesdropping and tampering, making it difficult for an attacker to impersonate either party and intercept sensitive information. With traditional client-server models, the failure or unavailability of a central authentication server can disrupt the entire authentication process. In a P2P cloud, the absence of a single central point of authentication means that the system can continue to function even if some nodes are temporarily unavailable.

5. Conclusion

The proposed authentication framework represents a significant advancement in enhancing data security within P2P cloud environments. By using Google Cloud Platform(GCP), integrating the SRP protocol and the AES algorithm, the system establishes a robust defense against potential vulnerabilities, ensuring the confidentiality of exchanged information. The SRP protocol's incorporation of cryptographic primitives bolsters mutual authentication between cloud servers and users. Leveraging a zero-knowledge proof mechanism during user registration, the protocol effectively neutralizes the risks stemming from password leaks. Moreover, the framework's emphasis on secure key exchange underscores the commitment to fostering confidential communication channels between peers. As the digital realm continues to evolve, solutions like these pave the way for safer, more secure data exchanges, driving the future of P2P cloud environments towards greater resilience and protection.

References

[1] Zhong, Hong, Chuanwang Zhang, Jie Cui, Yan Xu, and Lu Liu. "Authentication and key agreement based on anonymous identity for peer-to-peer cloud." IEEE transactions on cloud computing 10, no. 3 (2020): 1592-1603.

[2] Kaaniche, Nesrine. "Cloud data storage security based on cryptographic mechanisms." PhD diss., Evry, Institute national des telecommunications, 2014.

[3] Sarojini, G., A. Vijayakumar, and K. Selvamani. "Trusted and reputed services using enhanced mutual trusted and reputed access control algorithm in cloud." *Procedia Computer Science* 92 (2016): 506-512.

[4] Ali, Mazhar, Revathi Dhamotharan, Eraj Khan, Samee U. Khan, Athanasios V. Vasilakos, Keqin Li, and Albert Y. Zomaya. "SeDaSC: secure data sharing in clouds." *IEEE Systems Journal* 11, no. 2 (2015): 395-404.

[5] Jia, Xiaoying, Debiao He, Neeraj Kumar, and Kim-Kwang Raymond Choo. "A provably secure and efficient identity-based anonymous authentication scheme for mobile edge computing." *IEEE Systems Journal* 14, no. 1 (2019): 560-571.

[6] Cui, Jie, Xiaoyu Zhang, Hong Zhong, Jing Zhang, and Lu Liu. "Extensible conditional privacy protection authentication scheme for secure vehicular networks in a multi-cloud environment." *IEEE Transactions on Information Forensics and Security* 15 (2019): 1654-1667.

[7] Gupta, Ishu, Ashutosh Kumar Singh, Chung-Nan Lee, and Rajkumar Buyya. "Secure data storage and sharing techniques for data protection in cloud environments: A systematic review, analysis, and future directions." *IEEE Access* (2022).

[8] Zhang, Kai, Zhe Jiang, Jianting Ning, and Xinyi Huang. "Subversion- resistant and consistent attribute-based keyword search for secure cloud storage." *IEEE Transactions on Information Forensics and Security* 17 (2022): 1771-1784.

[9] Shabbir, Maryam, Ayesha Shabbir, Celestine Iwendi, Abdul Rehman Javed, Muhammad Rizwan, Norbert Herencsar, and Jerry Chun-Wei Lin. "Enhancing security of health information using modular encryption standard in mobile cloud computing." *IEEE Access* 9(2021): 8820-8834.

[10] Nhlabatsi, Armstrong, Jin B. Hong, Dong Seong Kim, Rachael Fernandez, Alaa Hussein, Noora Fetais, and Khaled M. Khan. "Threat- specific security risk evaluation in the cloud." *IEEE Transactions on Cloud Computing* 9, no. 2 (2018): 793-806.