International Journal of Interdisciplinary Innovative Research &Development (IJIIRD)
ISSN: 2456-236X
Vol. 08 Special Issue 01 | 2023

Dual Access Control For Cloud-Based Data Storage And Sharing

Mr. Sujeet D. Ghule [1], Mr. Sunny V. Pawar [2], Mr. Santosh R. Nikhade [3], Mr. Prafful S Ambilkar[4], Prof. Nisha Robade [5]

[1 2 3 4 5] Department of Computer Science & Engineering, Dr. V. B. kolte College of Engineering, Malkapur, Maharashtra, India

ABSTRACT

Cloud-based data storage services have gained attention in recent years because of their efficient and cost-effective management. However, as these services are provided over an open network, service providers need to implement secure data storage and sharing mechanisms to safeguard the privacy and confidentiality of their users' data. Although encryption is a common technique to secure sensitive data, it is insufficient for practical data management. Therefore, it is crucial to have an effective access control mechanism that can manage download requests and prevent denial-of-sustainability attacks that may affect the service's accessibility for the users. This article discusses the concept of dual access control in cloud-based storage. It proposes two control mechanisms to regulate data access and download requests, respectively. These mechanisms are customized for different settings, and their efficiency and security are analyzed in detail. To ensure secure data storage and sharing, cloud-based data storage services must employ both encryption and access control mechanisms. Encryption provides an initial layer of protection against unauthorized access, while access control guarantees that only authorized users can access and download the data. Dual access control mechanisms allow service providers to effectively manage and secure their users' data while ensuring that their services remain accessible and reliable.

Keyword: ***- Include Cloud-based data sharing, cloud storage service, access control, attribute-based encryption, Intel SGX***

1. INTRODUCTION

Cloud-based storage services have gained significant attention from academia and industry in recent decades due to their numerous benefits, such as access flexibility and cost savings on local data management facilities. Many individuals and companies are opting to outsource their data to remote clouds to reduce the expenses associated with upgrading their data management infrastructure. However, security concerns over potential breaches of outsourced data are a significant obstacle that hinders the widespread adoption of cloud-based storage services. In many practical applications, outsourced data needs to be shared with others, which raises the risk of unauthorized access to sensitive data. To mitigate this risk, data encryption is recommended prior to uploading to the cloud. One potential solution is to use encryption techniques such as AES directly on the outsourced data before uploading to the cloud, which allows only specified cloud users with a valid decryption key to access the data [1].

For instance, Dropbox users may share photos with friends, but without encryption, sharing links can be visible within the Dropbox administration level, which poses a significant security risk. Data encryption provides an additional layer of security and privacy to the outsourced data, making it more challenging for unauthorized users to access sensitive information [1]. By using encryption techniques and access control mechanisms, cloud-based storage services can provide secure data storage and sharing while ensuring the privacy and confidentiality of the data. This article proposes a novel mechanism, referred to as dual access control, to address the two challenges mentioned above. To ensure the security of data in cloud-based storage services, attribute-based encryption (ABE) has emerged as a promising solution. ABE not only offers confidentiality for outsourced data but also enables fine-grained control over it. Specifically, Ciphertext-Policy ABE (CP-ABE) presents an effective data encryption method that allows access policies to be defined over encrypted data, specifying access privileges for potential data receivers. However, while

this paper employs CP-ABE, it is important to note that simply using this technique is not sufficient to create a sophisticated mechanism that can guarantee control over both data access and download requests.

2 BACKGROUND

2.1 Ciphertext-Policy Attribute-Based

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is a type of encryption scheme that allows access control based on attributes of users or data. In CP-ABE, access policies are expressed as a set of attributes that are required for decryption, and ciphertexts are encrypted under these access policies [2] .
Here are four algorithms commonly used in CP-ABE:

- **Key Generation Algorithm:** This algorithm is used by the authority to generate the public and private keys for the system. It takes as input the security parameters and generates the public parameters, master secret key, and a set of attributes.
- **Encryption Algorithm:** This algorithm is used by the data owner to encrypt data under a specific access policy. It takes as input the public parameters, the access policy, and the plaintext message, and outputs the ciphertext.
- **Attribute-Based Decryption Algorithm:** This algorithm is used by a user to decrypt a ciphertext that is encrypted under an access policy that matches the user's attributes. It takes as input the user's secret key, the ciphertext, and the public parameters, and outputs the plaintext message.
- **Revocation Algorithm:** This algorithm is used to revoke a user's access to the system. It takes as input the user's attributes and updates the public parameters and the user's secret key to ensure that the user can no longer decrypt ciphertexts encrypted under access policies that require the revoked attributes.

2.2 Ciphertext-Policy Attribute-Based Encryption

Authenticated Encryption with Associated Data (AEAD) is a cryptographic scheme that combines encryption and authentication to provide a secure and efficient method of protecting data. AEAD is used to encrypt data while also ensuring that the data has not been tampered with during transit or storage [3]. AEAD is commonly used in scenarios where data needs to be securely transmitted over an untrusted network or stored on an untrusted device. AEAD ensures that the data remains confidential, meaning that it cannot be read by unauthorized parties, and authentic, meaning that the data has not been tampered with or modified by unauthorized parties. AEAD combines two cryptographic primitives, encryption and message authentication code (MAC), into a single operation. Encryption is used to scramble the plaintext data so that it is unreadable by unauthorized parties [3]. A MAC is then computed over the ciphertext and any associated data to ensure that the ciphertext and associated data have not been tampered with or modified.
AEAD is designed to provide three key properties:

- **Confidentiality:** The plaintext data should not be revealed to unauthorized parties.
- **Integrity**: The ciphertext and associated data should not be tampered with or modified by unauthorized parties.
- **Authenticity**: The ciphertext and associated data should be authenticated so that the recipient can be sure that the data has not been tampered with or modified.

AEAD can be implemented using different cryptographic algorithms, such as AES (Advanced Encryption Standard) in combination with HMAC (Hash-based Message Authentication Code) [4]. Some of the commonly used AEAD algorithms are:

- AES-GCM (Galois/Counter Mode): AES-GCM is a widely used AEAD algorithm that uses a block cipher mode of operation to provide encryption and a universal hash function to provide authentication. It is used in many applications, including TLS and IPSec.
- ChaCha20-Poly1305: ChaCha20-Poly1305 is a AEAD algorithm that combines a stream cipher and a MAC to provide encryption and authentication. It is used in protocols like Google's QUIC and the latest version of TLS.
- AES-CCM (Counter with CBC-MAC): AES-CCM is another AEAD algorithm that uses AES in combination with a CBC-MAC to provide encryption and authentication. It is commonly used in IEEE 802.15.4 wireless sensor networks.
- AEAD is a cryptographic scheme that provides both confidentiality and authenticity of data. It is widely used in modern communication protocols to secure data transmission and storage. The implementation of AEAD

algorithms may vary, but they all follow the same basic principle of combining encryption and authentication to provide secure communication.

2.3 Intel SGX

Intel Software Guard Extensions (SGX) is a set of hardware-based security features built into modern Intel processors. SGX provides a trusted execution environment (TEE) in which an application can execute securely and protect its code and data from unauthorized access or modification by the operating system, hypervisor, or other applications on the same system [4]. SGX enables the creation of secure enclaves, which are regions of memory that are protected from external access by hardware-based security mechanisms. Enclaves are created and managed by trusted applications, which are responsible for ensuring that the code and data inside the enclave are protected from unauthorized access or modification.

SGX provides several key security features, including:

- Memory Isolation: Enclaves are isolated from the rest of the system's memory, including the operating system and other applications. This isolation is enforced by the hardware, which prevents any access to enclave memory from outside the enclave.
- Code and Data Protection: The code and data inside an enclave are encrypted and protected from unauthorized access or modification. Any attempts to access or modify the code or data from outside the enclave will result in an error.
- Attestation: SGX provides a mechanism for attesting to the integrity of an enclave. Enclaves can be attested to by trusted parties, such as remote servers, to ensure that the enclave has not been tampered with and is executing the correct code.

SGX can be used to build a wide range of secure applications, including secure enclaves for cloud computing, secure key management for encryption, and secure transaction processing for financial applications. SGX can also be used to protect sensitive data, such as personal information or intellectual property, from unauthorized access or theft.

Intel SGX is a hardware-based security feature that provides a trusted execution environment for applications to execute securely and protect their code and data from unauthorized access or modification. SGX provides a range of security features, including memory isolation, code and data protection, and attestation, making it a powerful tool for building secure applications.

3. SYSTEM ARCHITECTURE AND SECURITY MODEL

3.1 System Architecture

Figure 1 illustrates the architecture of our dual access control systems designed for cloud data sharing. The systems consist of several entities responsible for specific tasks. The Authority entity is responsible for initializing system parameters and registering data users. In the first proposed construction, it handles the call request from the cloud. The Data Owner entity holds the data and wants to outsource it to the cloud. They only want to share their data with specific users who satisfy certain conditions. Once their data is uploaded to the cloud, they will be offline [5].

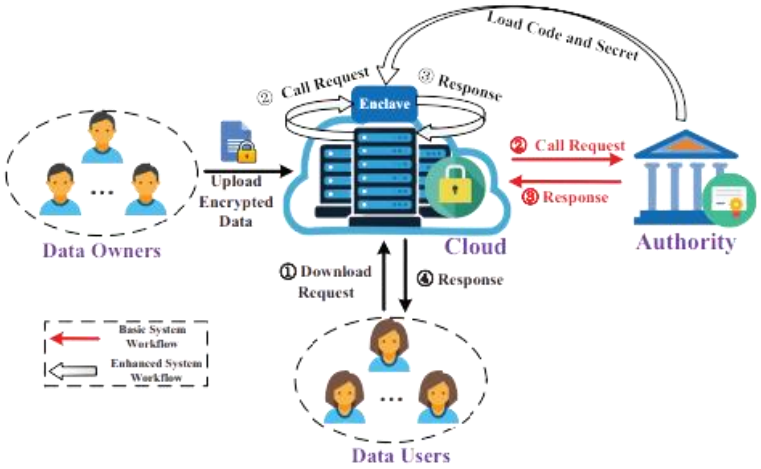

Fig -1 Overview of system architecture

The Data User entity wants to download and decrypt the encrypted data shared in the cloud, and only authorized users can access the plaintext by downloading the encrypted file. The Cloud entity provides convenient storage services for data owners and data users, stores the outsourced data, and handles download requests. In the second system, the Enclave entity handles the call request from the cloud. The workflow begins with data owners encrypting their data under their chosen access policies and uploading it to the cloud. Authorized data users can download the shared data by sending a download request to the cloud. Upon receiving the request, the cloud follows different procedures for the two systems [6].
For the basic system, the cloud sends a call request to the authority, and after receiving a response, it sends a response back to the data user. For the enhanced system, the cloud sends a call request to the enclave, and after receiving a response, it sends a response back to the data user. In both systems, the cloud ensures that only authorized users can access and download the shared data, thus providing a secure and efficient means of cloud data sharing [6] .

3.2 Security Assumptions

The security assumption of each entity is described as follows.
Authority: is fully trusted by other entities.
Data Owner: A data owner is deemed honest when they encrypt outsourced data and upload it to the cloud in an honest manner.
Data User: A data user may be considered malicious if they attempt to access a shared file that they do not have permission to access, potentially leading to the launching of EDoS attacks.
Cloud : The cloud is characterized as honest-but-curious since it has the ability to gather sensitive information through observation of the transcript, while adhering to the given specifications. The cloud stores outsourced data and manages access control on download requests honestly. However, it may try to deduce additional information that is not intended to be known beyond what is provided by the transcript.
Enclave : Enclave is a trusted entity that can execute loaded programs in an honest manner, even using secret data if necessary. The program and static data inside the enclave are not accessible or modifiable from the outside, even for root or any other special-access program. This hardware-based guarantee is provided by the Software Guard Extensions (SGX).

4 THE PROPOSED SYSTEMS

4.1 System Overview

Our proposed system for data protection is a hybrid system that combines the efficiency of a symmetric-key system with the convenience of a public-key system. We use the Key/Data Encapsulation Mechanism (KEM/DEM) setting and employ the CP-ABE technique as the basic building block to ensure anonymous data sharing, the confidentiality of shared data, and access control on shared data [7].
We introduce a new approach that allows the data owner to generate a download request containing a randomized form of the secret key. The download request retains the "decryption capability" of the secret key to test whether the data owner is authorized to decrypt the shared ciphertext(s). This approach avoids using the "testing" ciphertext and enables the cloud to check the authorization of the data owner anonymously without revealing any sensitive information. To prevent leaking secret information to the cloud, we use the help of the authority or the enclave of Intel SGX during the verification of the download request procedure[8].
Our first system involves the help of the authority during the verification of download requests, while the second system uses the enclave of Intel SGX and does not require the authority to be always online. Our technique is general and can be applied to most current CP-ABE constructions based on bilinear maps. By employing dual access control, our system provides strong security and privacy guarantees for shared data on the cloud, defending against EDoS attacks.

4.2 The Basic System

The procedures are described as follows.
Parameter Initialization :

The initial step in this process involves the setup of the system and the dissemination of public parameters that will be utilized by other entities. The authority is responsible for this task and executes the following actions: initialize a security parameter and acquire a bilinear map group system by calling the group generator G with G; GT as groups of prime order p, and e being a bilinear map. It is important to note that the generation of PK; MSK is almost

identical to the underlying, with the exception that MSK contains an additional parameter A. Therefore, the above procedure operates similarly to the algorithm Setup, except that it adds a to the master secret key MSK[10].
Data User Registration: As part of the Data User Registration process, the authority is responsible for providing each registered data user with a secret access credential. To gain entry into the system, each data user must register with the authority.
Shared File Generation and Outsourcing: This procedure involves data owners encrypting their data based on their chosen access policies and upload the encrypted data to the cloud.
Download Request Generation: To retrieve an encrypted file from the cloud, a data user initiates a download request.
Access Control on Download Request: The procedure of Access Control on Download Request is designed to ensure that only authorized data users can access the shared data through the download request.
Access Shared Data: During the Access, Shared Data procedure, a data user who has been granted authorization will utilize their secret access credential to decrypt an encrypted file that has been received, in order to gain access to the underlying data.

4. CONCLUSIONS

We have proposed two access control systems that effectively address long-standing issues in cloud-based data sharing, and have demonstrated their resistance to DDoS/EDoS attacks. Our approach for controlling download requests can be easily adapted to other CP-ABE systems. Through experimentation, we have found that the proposed systems have negligible computational and communication overheads compared to the underlying CP-ABE scheme. While our enhanced system takes advantage of the security features provided by enclaves, recent research has shown that enclaves may still leak information to malicious hosts through memory access patterns and other side-channel attacks. To address this, the concept of transparent enclave execution has been introduced, and we plan to explore the construction of a dual access control system for cloud data sharing using this model in future work.

5. REFERENCES

[1] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data.
[2] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization.
[3] Jianting Ning, Xiaolei Dong, Zhenfu Cao, Lifei Wei, and Xiaodong Lin. White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes.
[4] Gaurav Somani, Manoj Singh Gaur, and Dheeraj Sanghi. Ddos/edos attack in cloud: affecting everyone out there! In SIN 2015, pages 169–176. ACM, 2015.
[5] Ittai Anati, Shay Gueron, Simon Johnson, and Vincent Scarlata. Innovative technology for cpu based attestation and sealing. In Workshop on hardware and architectural support for security and privacy (HASP), volume 13, page 7. ACM New York, NY, USA, 2013.
[6] Alexandros Bakas and Antonis Michalas. Modern family: A revocable hybrid encryption scheme based on attribute-based encryption, symmetric searchable encryption and SGX. In SecureComm 2019, pages 472–486, 2019.
[7] Amos Beimel. Secure schemes for secret sharing and key distribution. PhD thesis, PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.
[8] John Bethencourt, Amit Sahai, and Brent Waters. Ciphertext-policy attribute-based encryption. In S&P 2007, pages 321–334. IEEE, 2007.
[9] Victor Costan and Srinivas Devadas. Intel sgx explained. IACR Cryptology ePrint Archive, 2016(086):1–118, 2016.
[10] Ben Fisch, Dhinakaran Vinayagamurthy, Dan Boneh, and Sergey Gorbunov. IRON: functional encryption using intel SGX. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, CCS 2017, pages 765–782, 2017.