DUAL ACCESS CONTROL FOR CLOUD-BASED DATA STORAGE AND SHARING

Bakkashetti Akshitha Sai[1], Dr.K.Bhargavi[2]
[1] PG Scholars, Department of CSE, **Teegala Krishna Reddy Engineering College,**Hyderabad, Telangana, India.
[2] Professor, Department of CSE, **Teegala Krishna Reddy Engineering College**, Hyderabad, Telangana, India.

ABSTRACT:

Cloud-based data storage service has drawn increasing interests from both academic and industry in the recent years due to its efficient and low cost management. Since it provides services in an open network, it is urgent for service providers to make use of secure data storage and sharing mechanism to ensure data confidentiality and service user privacy. To protect sensitive data from being compromised, the most widely used method is encryption. However, simply encrypting data (e.g., via AES) cannot fully address the practical need of data management. Besides, an effective access control over download request also needs to be considered so that Economic Denial of Sustainability (EDoS) attacks cannot be launched to hinder users from enjoying service. In this paper, we consider the dual access control, in the context of cloud-based storage, in the sense that we design a control mechanism over both data access and download request without loss of security and efficiency. Two dual access control systems are designed in this paper, where each of them is for a distinct designed setting. The security and experimental analysis for the systems are also presented

KEYWORDS: Searchable Encryption, Multi-Keyword Search, Multi-User Access, Search Pattern, Access Pattern.

DOI: 10.48047/ecb/2023.12.Si8.682

INTRODUCTION

IN the recent decades, cloud-based storage service has attracted considerable attention from both academia and industries. It may be widely used in many Internet-based

commercial applications (e.g., Apple iCould) due to its long-list benefits including access flexibility and free of local data management. Increasing number of individuals and companies nowadays prefer to outsource their data to remote cloud in such a way that they may reduce the cost of upgrading their local data management facilities/devices.

However, the worry of security breach over outsourced data may be one of the main obstacles hindering Internet users from widely using cloud-based storage service. In many practical applications, outsourced data may need to be further shared with others. For example, a Dropbox user Alice may share photos with her friends. Without using data encryption, prior to sharing the photos, Alice needs to generate a sharing link and further share the link with friends. Although guaranteeing some level of access control over unauthorized users (e.g., those are not Alice's friends), the sharing link may be visible within the Dropbox administration level (e.g., administrator could reach the link). Since the cloud (which is deployed in an open network) is not be fully trusted, it is generally recommended to encrypt the data prior to being uploaded to the cloud to ensure data security and privacy. One of the corresponding solutions is to directly employ an encryption technique(e.g., AES) on the outsourced data before uploading to cloud, so that only specified cloud user (with valid decryption key) can gain access to the data via valid decryption. To prevent shared photos being accessed by the "insiders" of the system, a straightforward way is to designate the group of authorized data users prior to encrypting the data. In some cases, nonetheless, Alice may have no idea about who the photo receivers/users are going to be. It is possible that Alice only has knowledge of attributes w.r.t. photo receivers. In this case, traditional public key encryption (e.g., Paillier Encryption), which requires the

encryptor to know who the data receiver is in advance, cannot be leveraged. Providing policy-based encryption mechanism over the outsourced photos is therefore desirable, so that Alice makes use of the mechanism to define access policy over the encrypted photos to guarantee only a group of authorized users is able to access the photos. In a cloud-based storage service, there exists a common attack that is well-known as resource-exhaustion attack. Since a (public) cloud may not have any control over download request(namely, a service user may send unlimited numbers of download request to cloud server), a malicious service user may launch the denial-of-service (DoS)/distributed denial-of-service (DDoS) attacks to consume the resource of cloud storage service server so that the cloud service could not be able to respond honest users' service requests. As a result, in the "pay-as-you-go" model, economic aspects could be disrupted due to higher resource usage. The costs of cloud service users will rise dramatically as the attacks

In this paper, we propose a new mechanism, dubbed dual access control, to tackle the above aforementioned two problems. To secure data in cloud-based storage service, attribute-based encryption (ABE) [9] is one of the promising candidates that enables the confidentiality of out sourced data as well as fine-grained control over the outsourced data. In particular, Ciphertext-Policy ABE (CP-ABE) [5] provides an effective way of data encryption such that access policies, defining the access privilege of potential data receivers, can be specified over encrypted data. Note that we consider the use of CP-ABE in our mechanism in this paper. Nevertheless, simply employing CP-ABE technique is not sufficient to design an elegant mechanism guaranteeing the control of both data access and download request.

LITERATURE SURVEY

[1] 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.

Secure cloud storage is considered as one of the most important issues that both businesses and end-users take into account before moving their private data to the cloud. Lately, we have seen some interesting approaches that are based either on the promising concept of Symmetric Searchable Encryption (SSE) or on the well-studied field of Attribute-Based Encryption (ABE). In this paper, we propose a hybrid encryption scheme that combines both SSE and ABE by utilizing the advantages of both these techniques. In contrast to many approaches, we design a revocation mechanism that is completely separated from the ABE scheme and solely based on the functionality offered by SGX

[2] Antonis Michalas. The lord of the shares: combining attributebased encryption and searchable encryption for flexible data sharing. In SAC 2019, pages 146–155, 2019

Secure cloud storage is considered one of the most important issues that both businesses and end-users are considering before moving their private data to the cloud. Lately, we have seen some interesting approaches that are based either on the promising concept of Symmetric Searchable Encryption (SSE) or on the well-studied field of Attribute-Based Encryption (ABE). In the first case, researchers are trying to design protocols where users' data will be protected from both internal and external attacks without paying the necessary attention to the problem of user revocation. On the other hand, in the second case existing approaches address the problem of revocation.

However, the overall efficiency of these systems is compromised since the proposed protocols are solely based on ABE schemes and the size of the produced ciphertexts and the time required to decrypt grows with the complexity of the access formula. In this paper, we propose a protocol that combines both SSE and ABE in a way that the main advantages of each scheme are used. The proposed protocol allows users to directly search over encrypted data by using an SSE scheme while the corresponding symmetric key that is needed for the decryption is protected via a Ciphertext-Policy Attribute-Based Encryption scheme.

[3] G. Wang, C. Liu, Y. Dong, P. Han, H. Pan, and B. Fang, "Idcrypt: A multi-user searchable symmetric encryption scheme for cloud applications," IEEE Access, vol. 6, pp. 2908–2921, 2018.

Searchable Encryption (SE) has been extensively examined by both academic and industry researchers. While many academic SE schemes show provable security, they usually expose some query information (e.g., search and access patterns) to achieve high efficiency. However, several inference attacks have exploited such leakage, e.g., a query recovery attack can convert opaque query trapdoors to their corresponding keywords based on some prior knowledge. On the other hand, many proposed SE schemes require significant modification of existing applications, which makes them less practical, weak in usability, and difficult to deploy. In this paper, we introduce a secure and practical searchable symmetric encryption scheme with provable security strength for cloud applications, called IDCrypt, which improves the search efficiency, and enhances the security strength of SE using symmetric cryptography. We further point out the main challenges in securely searching on multiple indexes and sharing encrypted data between multiple users. To address the above issues, we propose a token-adjustment search scheme to preserve

the search functionality among multi-indexes, and a key sharing scheme which combines identity-based encryption and public-key encryption. Our experimental results show that the overhead of the key sharing scheme is fairly low.

[4] Kaiping Xue, Weikeng Chen, Wei Li, Jianan Hong, and Peilin Hong. Combining data owner-side and cloud-side access control for encrypted cloud storage. IEEE Transactions on Information Forensics and Security, 2018.

People endorse the great power of cloud computing, but cannot fully trust the cloud providers to host privacy-sensitive data, due to the absence of user-to-cloud controllability. To ensure confidentiality, data owners outsource encrypted data instead of plaintexts. To share the encrypted files with other users, ciphertext-policy attribute-based encryption (CP-ABE) can be utilized to conduct fine-grained and owner-centric access control. But this does not sufficiently become secure against other attacks. Many previous schemes did not grant the cloud provider the capability to verify whether a downloader can decrypt. Therefore, these files should be available to everyone accessible to the cloud storage. A malicious attacker can download thousands of files to launch economic denial of sustainability (EDoS) attacks, which will largely consume the cloud resource. The payer of the cloud service bears the expense. Besides, the cloud provider serves both as the accountant and the payee of resource consumption fee, lacking the transparency to data owners. These concerns should be resolved in real-world public cloud storage. In this paper, we propose a solution to secure encrypted cloud storages from EDoS attacks and provide resource consumption accountability. It uses CP-ABE schemes in a black-box manner and complies with arbitrary access policy of the CP-ABE. We present two protocols for different

settings, followed by performance and security analysis

[5] Jianting Ning, Zhenfu Cao, Xiaolei Dong, Kaitai Liang, Hui Ma, and Lifei Wei. Auditable σ-time outsourced attribute-based encryption for access control in cloud computing. IEEE Transactions on Information Forensics and Security, 13(1):94–105, 2018..

As a sophisticated mechanism for secure finegrained access control over encrypted data, ciphertext-policy attribute-based encryption (CP-ABE) is one of the highly promising candidates for cloud computing applications. However, there exist two main long-lasting open problems of CP-ABE that may limit its wide deployment in commercial applications. One is that decryption yields expensive pairing cost which often grows with the increase of access policy size. The other is that one is granted access privilege for unlimited times as long as his attribute set satisfies the access policy of a given ciphertext. Such powerful access rights, which are provided by CP-ABE, may be undesirable in real-world applications (e.g., pay-as-youuse). To address the above drawbacks, in this paper, we propose a new notion called auditable σ-time outsourced CF-ABE, which is believed to be applicable to cloud computing. In our notion, expensive pairing operation incurred by decryption is offloaded to cloud and meanwhile, the correctness of the operation can be audited efficiently. Moreover, the notion provides σ-time fine-grained access control. The cloud service provider may limit a particular set of users to enjoy access privilege for at most σ times within a specified period. As of independent interest, the notion also captures key-leakage resistance. The leakage of a user's decryption key does not help a malicious third party in decrypting the ciphertexts belonging to the user. We design a concrete construction (satisfying our notion) in the key encapsulation mechanism setting

based on Rouselakis and Waters (prime order) CP-ABE, and further present security and extensive experimental analysis to highlight the scalability and efficiency of our construction

EXISTING SYSTEM:

The existing works, by using normal servers for storing and sharing data that causes un security lack of privacy. There is a chance of stole our data this is the main drawback of existing system to overcome this difficulty we can go for proposed system.

DISADVANTAGES:

- Searching the stored documents takes time linear in the size of the database
- It uses heavy arithmetic operations
- Less security.

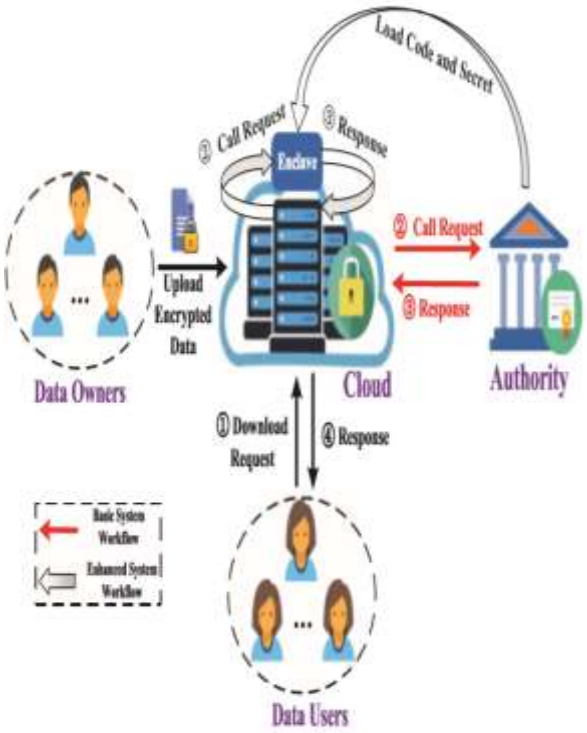
PROPOSED SYSTEM:

In proposed system, we propose a new mechanism, dubbed dual access control, to tackle the above aforementioned two problems. To secure data in cloud-based storage service, attribute-based encryption is one of the promising candidates that enables the confidentiality of out sourced data as well as fine-grained control over the outsourced data

ADVANTAGES:

- Provides more security.
- It uses simple arithmetic operations.
- Storage capacity is high.

BLOCK DIAGRAM:



SYSTEM SPECIFICATIONS:

H/W CONFIGURATION:

- **Processor - I3/Intel Processor**
- Hard Disk - 160GB
- Key Board - Standard Windows Keyboard
- Mouse - Two or Three Button Mouse
- Monitor - SVGA

S/W CONFIGURATION:

- Operating System : Windows 7/8/10
- IDE : Pycharm
- Server side scripts : HTML, CSS, Js
- Libraries Used : Pandas, smtplib, Flask
- Technology : Python 3.6+

MODULES:

1. DATA OWNER:

Register:

Data owner can Register and login with valid credentials

Upload File:

Data provider can upload the file.

View File:

Data Owner can view uploaded file once means whether the file is correctly uploaded or not.

2. USER:

Register:

Data user can do registration with his details.

Login:

The user needs to register and the data stored in MySQL database.

Search a File:

Data user can search a file based on the keyword ,if file is available then user can view file and send request to cloud to download the file.

Get Key & Download

Once User Request can accept get the key to cloud provider user can download the file.

3.Cloud Provider

Login

Cloud provider can login with his/her credentials.

View Files:

Cloud can view all uploaded files.

View Users:

Cloud can view all the users details to give permission for login the website.

View Data Providers:

Cloud can view all the data providers details to give permission for login the website.

Send Key request to authority:

Cloud gets a key from authority and send to the authority.

4. Authority:

Login:

Authority login and view users and give authorization to users.

Generate key to users:

Authority generate key to users.

ALGORITHM:

The encryption process uses a set of specially derived keys called round keys. These are applied, along with other operations, on an array of data that holds exactly one block of data?

the data to be encrypted. This array we call the state array.

You take the following aes steps of encryption for a 128-bit block:

Derive the set of round keys from the cipher key.

Initialize the state array with the block data (plaintext).

Add the initial round key to the starting state array.

Perform nine rounds of state manipulation.

Perform the tenth and final round of state manipulation.

Copy the final state array out as the encrypted data (ciphertext).

The reason that the rounds have been listed as "nine followed by a final tenth round" is because the tenth round involves a slightly different manipulation from the others.

The block to be encrypted is just a sequence of 128 bits. AES works with byte quantities so we first convert the 128 bits into 16 bytes. We say "convert," but, in reality, it is almost certainly stored this way already. Operations in RSN/AES are performed on a two-dimensional byte array of four rows and four columns. At the start of the encryption, the 16 bytes of data.

STEPS FOR EXECUTING THE PROJECTS

1. Install the required packages

2. Defining the database connections.

3. Creating all the data base tables which we use in this project.

4. Remodel the CSS & HTML pages.

5. Build the project based on the modules.

6. Run → Run Python file(app.py) →Copy the url link → Paste at any browser → then follow below screen shots.

SCREENSHOT

Home page:

Cloud service provider:

Cloud service provider home page:

View owners and authorization:

View Users:

Authority:

View Users Request:

Authority Home page:

View Users requests and Generating key:

Dataowner registration page:

View Users for authorization:

Dataowner loginpage:

Dataowners Home:

View Files:

Upload File:

View All Files and request dual access:

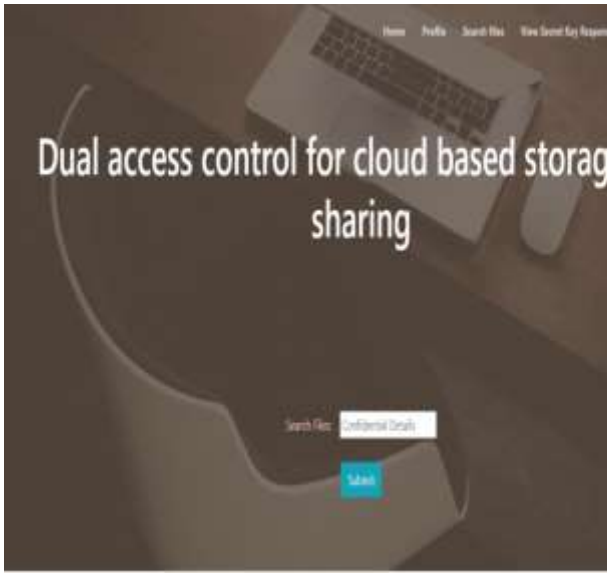
View Dual access response:

Datauser registration page;

View Dual access response;

Datauser login page:

Search files:



View Secret key response:

CONCLUSION:

In this paper. We addressed an interesting and long-lasting problem in cloud-based data sharing, and presented two dual access control systems. The proposed systems are resistant to DDoS/EDoS attacks. We state that the technique used to achieve the feature of control on download requestis "transplantable" toother CP-ABEconstructions. Ourexperimental results show that the proposed systems do not impose any significant computational and communication overhead (compared to its underlying CP-ABE building block). In our enhanced system, we employ the fact that the secret information loaded into the enclave cannot be extracted. However, recent work shows that enclave may leak some amount so fits secret(s) to a malicious host through the memory access patterns or other related side-channel attacks. The model of transparent enclave execution is

hence introduced in. Constructing a dual access control system for cloud data sharing from transparent enclave is an interesting problem. In our future work ,we will consider the corresponding solution to the problem

REFERENCES

[1] Joseph A Akinyele, Christina Garman, Ian Miers, Matthew W Pagano, Michael Rushanan, Matthew Green, and Aviel D Rubin. Charm: a framework for rapidly prototyping cryptosystems. Journal of Cryptographic Engineering, 3(2):111–128, 2013.

[2] 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.

[3] 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.

[4] Amos Beimel. Secure schemes for secret sharing and key distribution. PhD thesis, PhD thesis, Israel Institute of Technology, Technion, Haifa, Israel, 1996.

[5] JohnBethencourt,AmitSahai,andBrent Waters. Ciphertext-policy attribute-based encryption. In S&P 2007, pages 321–334. IEEE, 2007.

[6] Victor Costan and Srinivas Devadas. Intel sgx explained. IACR Cryptology ePrint Archive, 2016(086):1–118, 2016.

[7] 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.

[8] Eiichiro Fujisaki and Tatsuaki Okamoto. Secure integration of asymmetric and symmetric encryption schemes. In Advances in Cryptology-CRYPTO 1999, pages 537–554. Springer, 1999.

[9] Vipul Goyal, Omkant Pandey, Amit Sahai, and Brent Waters. Attribute-based encryption for fine-grained access control of encrypted data. In ACM CCS 2006, pages 89–98. ACM, 2006.

[10] Jinguang Han, Willy Susilo, Yi Mu, Jianying Zhou, and Man Ho Allen Au. Improving privacy and security in decentralized ciphertext-policy attribute-based encryption. IEEE transactions on information forensics and security, 10(3):665–678, 2015.

[11] Christofer Hoff. Cloud computing security: From ddos (distributed denial of service) to edos (economic denial of sustainability). http://www. rationalsurvivability.com/blog/?p=66.

[12] Joseph Idziorek, Mark Tannian, and Doug Jacobson. Attribution of fraudulent resource consumption in the cloud. In IEEE CLOUD 2012, pages 99–106. IEEE, 2012.

[13] Simon Johnson, Vinnie Scarlata, Carlos Rozas, Ernie Brickell, and Frank Mckeen. Intel R software guard extensions: Epid provision-ing and attestation services. White Paper, 1:1–10, 2016.

[14] Sangho Lee, Ming-Wei Shih, Prasun Gera, Taesoo Kim, Hyesoon Kim, and Marcus Peinado. Inferring fine-grained control flow inside sgx enclaves with branch shadowing. In 26th USENIX Security Symposium, USENIX Security, pages 16–18, 2017.

[15] Jiguo Li, Xiaonan Lin, Yichen Zhang, and Jinguang Han. Ksfoabe: outsourced attribute-based encryption with keyword search functionforcloudstorage. IEEETransactionsonServicesComputing, 10(5):715–725, 2017.