DUAL ACCESS CONTROL MECHANISMS FOR CLOUD DATA STORAGE AND SHARING

Dr. Mohd Azeemullah[1], Dr. Vijay Pal Singh[2]
[1]Assistant professor, Department of CSE, Sphoorthy Engineering College, Nadergul, Hyderabad
azeemullah889@gmail.com
[2]Associate Professor and Dean, Dept. of CSE, OPJS University, Churu.
vptilotiya@gmail.com

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.

1. 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 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 scale up. This has been known as Economic Denial of Sustainability (EDoS) attack [32], [33], which targets to the cloud adopter's economic resources. Apart from economic loss, unlimited download itself could open a window for network attackers to observe the encrypted download data that may lead to some potential information leakage (e.g., file size). Therefore, an effective control over download request for outsourced (encrypted) data is also needed.

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 outsourced 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.

A strawman solution to the control of download request is to leverage dummy ciphertexts to verify data receiver's decryption rights. It, concretely, requires data owner, say Alice, to upload multiple "testing" ciphertexts along with the "real" encryption of data to cloud, where the "testing" ciphertexts are the encryptions of dummy messages under the same access policy as that of the "real" data. After receiving a download request from a user, say Bob, cloud asks Bob to randomly decrypt one of the "testing" ciphertexts. If a correct result/decryption is returned (i.e. indicating Bob is with valid decryption rights), Bob is authorized by Alice to access the "real" data, so that the cloud allows Bob to download the corresponding ciphertext.

Nevertheless, several disadvantages of the above approach may be identified as follows. First of all, the data owner, Alice, is required to encrypt a number of dummy ciphertexts under the same policy as the "real" ciphertext. This may yield a considerable computational overhead for Alice, which may bring inconvenience in practice, for example, Alice just wants to upload one photo to iCloud from her cellphone, but needs to prepare more than one ciphertexts.

Second, all ciphertexts, including dummy ones, are uploaded to cloud at the same time. This inevitably imposes extra cost on network bandwidth (as well as prolonging data uploading time), which may not be applicable to some service users whose cellular network is under pay-as-you-go plan or equipped with old generation of broadband cellular network technology (e.g., 3G). Third, a data receiver/user, Bob, has to additionally decrypt a random-chosen "testing" ciphertext from cloud, as a test of his valid download request. As a result, Bob has to "pay" double (decryption price) for accessing to the "real" data, which again may not be scalable in resource constrained setting.

1.1 Our Results and Contributions

We answer the aforementioned question affirmatively by presenting two secure and efficient cloud-based dual access control systems1 in different contexts. With the aim of providing an efficient way of dual access control, we briefly introduce the technical roadmap as follows. To guarantee the confidentiality of outsourced data without loss of policy-based access control, we start with a CP-ABE system [36], which is seen as one of the building blocks. We further employ an effective control over data users' download request on the top of the CP-ABE system. We design a new approach to avoid using the technique of "testing" ciphertext. Specifically, we allow data user to generate a download request. Upon receiving the download request, with help of the authority or the enclave of Intel SGX, a cloud server is able to check if the data user is authorized to gain access to the data. No other information is revealed to the cloud server except the knowledge of whether the user is authorized. Based on the above mechanism, the cloud maintains the control of the download request. The systems we propose are with the following distinct features:

(1) Confidentiality of Outsourced Data: In our proposed systems, the outsourced data is encrypted prior to being uploaded to cloud. No one can access them without valid access rights.

(2) Anonymity of Data Sharing: Given an outsourced data, cloud server cannot identify data owner, so that the anonymity of owner can be guaranteed in data storage and sharing.

(3) Fine-Grained Access Control Over Outsourced (Encrypted) Data: Data owner keeps controlling his encrypted data via access policy after uploading the data to cloud. In particular, a data owner can encrypt his outsourced data under a specified access policy such that only a group of authorized data users, matching the access policy, can access the data.

(4) Control Over Anonymous Download Request and Edos Attacks Resistance: A cloud server is able to control the download request issued by any system user, where the download request can set to be anonymous. With the control over download request, we state that our systems are resistant to EDoS attacks.

(5) High Efficiency: Our proposed systems are built on the top of the CP-ABE system [36]. Compared with [36], they do not incur significant additional computation and communication overhead. This makes the systems feasible for real-world applications.

The rest of the paper is organized as follows. In section 2, we review the related work which deals and the application security. We present the proposed system in section 3. We test the proposed system method in section 4. We analysis the system results in section 5. We conclude the paper in section 6.

2. RELATED WORK

Alexandros Bakas and Antonis Michalas, 2019. Secure cloud storage is considered as one of the most important issues that both businesses and end-users consider 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. 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 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.

Cloud computing plays a significant role in our daily routine. From casual internet users, to big corporations, the cloud has become an integral part of our lives. However, using services that are hosted and controlled by third parties raises several security and privacy concerns. Additionally, it has been observed that the number of attacks that target users' privacy has grown significantly. For example, in [14] it is stated that there has been a 300% increase in Microsoft cloud-based user's account attacks over the past couple of years. However, when considering a cloud-based environment, cyber-attacks performed by remote adversaries is only a part of the problem.

Antonis Michalas, 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.

Jianting Ning, Zhenfu Cao, Xiaolei Dong, Kaitai Liang, Hui Ma, and Lifei Wei, 2018. As a sophisticated mechanism for secure fine-grained 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.

As a new prevalent commercial paradigm, cloud computing has attracted attention from both academic and industrial communities. Thanks to the advanced development of cloud computing, many enterprises and individuals are allowed to outsource the considerable amount of data to cloud instead of building and maintaining local data centers. The cloud users may also enjoy various types of computing services offered by public cloud. Despite providing long-list advantages, cloud computing may yield the concerns on data security and privacy which hinder its widely use. Attribute-based encryption (ABE) is designed to protect the confidentiality of sensitive data and further to provide fine-grained access control. It may become one of the promising candidates to address the security concern in cloud computing. In particular, ciphertext-policy ABE (CP-ABE) provides a scalable way of encrypting data such that enterprises and individuals can specify access policies over attributes that the potential data users possess. The users are authorized to decrypt respective pieces of outsourced data if their attribute sets satisfy the specified access policies. However, there still exist two major long-lasting problems that hinder the wide-range deployment and commercial applications of CP-ABE to date.

Jianting Ning, Zhenfu Cao, Xiaolei Dong, and LifeiWei, 2018. Ciphertext-policy attribute-based encryption (CP-ABE) has been proposed to enable fine-grained access control on encrypted data for cloud storage service. In the context of CP-ABE, since the decryption privilege is shared by multiple users who have the same attributes, it is difficult to identify the original key owner when given an exposed key. This leaves the malicious cloud users a chance to leak their access credentials to outsourced data in clouds for profits without the risk of being caught, which severely damages data security. To address this problem, we add the property of traceability to the conventional CP-ABE. To catch people leaking their access credentials to outsourced data in clouds for profits effectively, in this paper, we first propose two kinds of non-interactive commitments for traitor tracing. Then we present a fully secure traceable CP-ABE system for cloud storage service from the proposed commitment. Our proposed commitments for traitor tracing may be of independent interest, as they are both pairing-friendly and homomorphic. We also provide extensive experimental results to confirm the feasibility and efficiency of the proposed solution.

Emerging cloud computing replaces traditional outsourcing techniques and provides an efficient and cost-effective mechanism for organizations and individuals to enforce highly scalable and technology-enabled management on their data. As a new commercial and exciting paradigm, it has attracted much attention from both industrial and academic world. Cloud storage service enables cloud users to outsource their data to the cloud so that themselves or other authorized users can access the outsourced cloud data anywhere and anytime. Despite lots of benefits provided by cloud storage service, the concerns on data security are believed the main obstacles hindering the wide usage of cloud storage service. Cloud users may worry about the privacy of their outsourced data due to some unauthorized access to outsourced data (such as the loss of physical control of outsourced data, etc.). To address the data security concerns, encryption has been applied on the data before outsourcing. Meanwhile, in many cases, cloud users may want to share their outsourced data to some potential users without knowing who will receive it. Thus, a fine-grained access control over outsourced data is desired. Attribute-Based Encryption (ABE, [15]) is a highly promising approach to protect outsourced data and provide fine-grained access control for cloud storage service. In the context of CP-ABE, ciphertexts and keys are labeled with access policies and sets of descriptive attributes respectively. In a CP-ABE system for cloud storage service, data owners can specify access policies over attributes that the potential authorized cloud users should possess, and those cloud users who are authorized will be issued access credentials corresponding to their attribute sets and can get access to the outsourced data. A cloud user is authorized if the set of attributes he/she possesses satisfies the access policy specified by data owners. Intuitively, CP-ABE not only enables fine-grained access control over outsourced data in clouds, but also providing a reliable method to protect the outsourced data in clouds.

Kaiping Xue, Weikeng Chen, Wei Li, Jianan Hong, and Peilin Hong, 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.

Florian Tramer, Fan Zhang, Huang Lin, Jean-Pierre Hubaux, Ari Juels, and Elaine Shi, 2017. Itecture extension, aim to provide strong confidentiality and integrity assurances for applications. Recent work, however, raises serious concerns about the vulnerability of such systems to side-channel attacks. We propose, formalize, and explore a cryptographic primitive called a Sealed-Glass Proof (SGP) that models computation possible in an isolated execution environment with unbounded leakage, and thus in the face of arbitrary side-channels. A SGP specifically models the capabilities of trusted hardware that can attest to correct execution of a piece of code, but whose execution is transparent, meaning that an application's secrets and state are visible to other processes on the same host. Despite this strong threat model, we show that SGPs enable a range of practical applications. Our key observation is that SGPs permit safe verifiable computing in zero-knowledge, as data leakage results only in the prover learning her own secrets. Among other applications, we describe the implementation of an end-to-end bug bounty (or zero-day solicitation) platform that couples a SGX-based SGP with a smart contract. Our platform enables a marketplace that achieves fair exchange, protects against unfair bounty withdrawals, and resists denial-of-service attacks by dishonest sellers. We also consider a slight relaxation of the SGP model that permits black-box modules instantiating minimal, side-channel resistant primitives, yielding a still broader range of applications. Our work shows how trusted hardware systems such as SGX can support trustworthy applications even in the presence of side channels.

Despite this strong threat model, we show that SGPs enable a range of practical applications. Our key observation is that SGPs permit safe verifiable computing in zero-knowledge, as data leakage results only in the prover learning her own secrets. Among other applications, we describe the implementation of an end-to-end bug bounty (or zero-day solicitation) platform that couples a SGX-based SGP with a smart contract. Our platform enables a marketplace that achieves fair exchange, protects against unfair bounty withdrawals, and resists denial-of-service attacks by dishonest sellers. We also consider a slight relaxation of the SGP model that permits black-box modules instantiating minimal, side-channel resistant primitives, yielding a still broader range of applications. Our work shows how trusted hardware systems such as SGX can support trustworthy applications even in the presence of side channels.

Trusted hardware platforms aim at creating isolated software execution environments that could lead to many practical applications of secure multiparty computation. For instance, Intel's newly released Software Guard Extensions [1] (SGX) let developers create secure enclaves that execute in isolation from the rest of a host's software, including its OS.

While these trusted platforms aim to protect the integrity, authenticity and confidentiality of enclaved programs against a variety of software or physical attacks, the confidentiality goal appears elusive. Recent work shows that enclaves may leak large amounts of sensitive information to a malicious host through their memory access patterns. Additional software or physical side-channel could further compromise an enclave's secrets. Such attacks are not unique to SGX, and a number of defences have been developed over the years, usually to protect highly sensitive cryptographic code. It is reasonable to assume that trusted hardware platforms can apply these methods to achieve strong protection for a limited set of cryptographic operations using long term secrets (e.g., cryptographic keys used for attesting to an enclave's contents or for protecting enclave memory). Such protection is essential: A successful extraction of a platform's private key would essentially translate into a total security break. A much more challenging goal is extending side-channel protections to arbitrary computations as the resulting performance degradation is non-negligible and it remains unclear whether data leaks can truly be fully eliminated.

3. METHODOLOGIES

Intel SGX

Intel Software Guard Extensions (SGX) is a set of new instructions available on recent-model Intel CPUs that allow for the creation of isolated execution environments called enclaves [19]. Our systems build on the notion of enclave, which is designed to run code and handle secrets in a trustworthy manner, even on a host where the system memory and OS are untrusted. The enclave provides three main security properties: isolation, sealing, and attestation. Isolation restricts access to a hardware guarded area of memory such that only that particular enclave can access it. Any other process on the same processor, even the OS, hypervisor, cannot access that memory. Sealing provides a way of encrypting enclave secrets for persistent storage to disk such that the secrets can be retrieved even if the enclave is torn down. Encryption is performed using a private seal key that is unique to that particular enclave, no process other than the exact same enclave can decrypt (or modify) it.

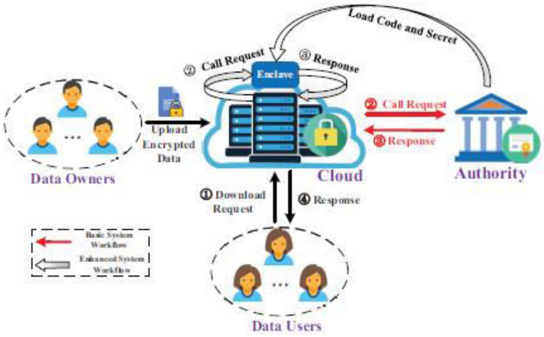


Fig. 1 Overview of system architecture

Attestation enables an entity to verify that the desired code is indeed running securely and unmodified within the enclave. In particular, there are two forms of attestation: local attestation and remote attestation [7]. Local attestation is used for attestation between two enclaves on the same platform. The two enclaves on the same machine can derive a shared key, called Report Key, using the Root Seal Key shared between them. Remote attestation enables an enclave to generate a report that can be verified by any remote entity. Specifically, in order to generate a quote, an enclave first attests to a special enclave called the Quoting Enclave locally and sends it a report. After verifying the received report, the Quoting Enclave converts it into a quote, which contains the same underlying data. Essentially, the quote is signed with a secret key for an anonymous group signature scheme called Intel Enhanced Privacy ID (EPID) [7], [13]. The signature generated from EPID can be essentially verified by using the group public key.

4. SYSTEM ARCHITECTURE AND SECURITY MODEL

4.1 System Architecture

The architectures of our dual access control systems for cloud data sharing are shown in Fig. 1. Concretely, the systems consist of the following entities:

- **Authority** is responsible for initializing system parameters and data user registration. Also, it handles the call request from the cloud in the first proposed construction.
- **Data owner** holds the data and wants to outsource his data to the cloud. In particular, data owners (only) want to share their data with those who satisfy certain conditions (e.g., professors or associate professors). They will be offline once their data have been uploaded to the cloud.
- **Data user** wants to download and decrypt the encrypted data shared in the cloud. Those who are authorized can download the encrypted file and further decrypt it to access the plaintext.
- **Cloud provides** convenient storage service for data owners and data users. Specifically, it stores the outsourced data from data users and handles the download requests sent by data users.
- **Enclave** handles the call request from the cloud (used in the second system).

Data owners encrypt their data under the access policies chosen by themselves and upload the encrypted data to the cloud. Authorized data users can download the shared data by sending a download request to the cloud. Upon receiving a download request from an authorized data user (see □ in Fig. 1), the cloud does as follows.

(a) For our basic system, the cloud sends a call request to the authority (see red □ between the cloud and the authority in Fig. 1). After receiving a response from the authority (see red □ between the cloud and the authority in Fig. 1), the cloud sends a response back to the data user (see □ in Fig. 1).

(b) For our enhanced system, the cloud sends a call request to the enclave (see black □ above the cloud in Fig. 1). After receiving a response from the enclave (see black□ above the cloud in Fig. 1), the cloud sends a response back to the data user.

3.2 Security Assumptions

The security assumption of each entity is described as follows.

- **Authority** is fully trusted by other entities.
- **Data owner** is honest in the sense that she/he encrypts the outsourced data and uploads the encrypted data to the cloud honestly.
- **Data user** is malicious in the sense that she/he may try to download the shared file which is not authorized for her/him and launch the EDoS attacks.
- **Cloud** is honest-but-curious in the sense that it may gather sensitive information curiously by observing the transcript but will not deviate from the specification. Specifically, it will store the outsourced data and handles the access control on the download request honestly. However, it may try to infer more information (they are not supposed to know) than what is revealed by the transcript.
- **Enclave** is fully trusted in the sense that it will execute the loaded program (using the loaded secret data inside if necessary) honestly2. In particular, the program and static data inside the enclave cannot be read or modified from the outside, even for root nor any other type of special-access program. It is a hardware-based guarantee provided by the Software Guard extensions (SGX).

4.2 Security Assumptions

The security assumption of each entity is described as follows.

- **Authority** is fully trusted by other entities.
- **Data owner** is honest in the sense that she/he encrypts the outsourced data and uploads the encrypted data to the cloud honestly.
- **Data user** is malicious in the sense that she/he may try to download the shared file which is not authorized for her/him and launch the EDoS attacks.
- **Cloud** is honest-but-curious in the sense that it may gather sensitive information curiously by observing the transcript but will not deviate from the specification. Specifically, it will store the outsourced data and handles the access control on the download request honestly. However, it may try to infer more information (they are not supposed to know) than what is revealed by the transcript.
- **Enclave** is fully trusted in the sense that it will execute the loaded program (using the loaded secret data inside if necessary) honestly2. In particular, the program and static data inside the enclave cannot be read or modified from the outside, even for root nor any other type of special-access program. It is a hardware-based guarantee provided by the Software Guard extensions (SGX).
- **Anonymous data sharing**: The identity of the data owner should not be public. In particular, for a newly uploaded file, the real identity of the file's owner cannot be identified by the cloud.
- **Confidentiality of shared data:** The data outsourced to the cloud should be invisible to the cloud and unauthorized data users.
- **Anonymous download request:** For a download re-quest sent from a data user, the request should be anonymous in the sense that the cloud cannot identify who sends this request.
- **Access control on download request:** To thwart a malicious data user's EDoS attacks, the shared data in the cloud can only be download by those who are authorized.
- **Access control on shared data:** The shared data can only be decrypted by those who are authorized. Based on the security assumptions and design goals. presented above, the security requirements of our systems include:
- Security against honest-but-curious cloud: a) The cloud cannot identify the owner of any newly uploaded file; b) The cloud cannot obtain the plaintext of the encrypted data stored on it; c) The cloud cannot identify the sender of any download request.
- Security against malicious data user: a) Any unauthorized data user cannot download the shared file(s)(i.e., resistant to data user's EDoS attacks); b) Any unauthorized data user cannot decrypt the shared file if the data user obtains the file. A data user is defined to be unauthorized if his/her attribute set does not satisfy the access policy of shared file.

5. PROPOSED ALGORITHM

Ciphertext-Policy Attribute-based-Encryption

Ciphertext-Policy Attribute-based-Encryption (CP-ABE) is a versatile encryption supporting fine-grained access control over encrypted data. In a CP-ABE system, each data user is issued with a secret key according to his attributes. A data owner can choose an access structure A and encrypt his data under A. The encrypted file can be decrypted by any data user whose attribute set satisfies A. CP-ABE systems proposed in recent years usually make essential use of linear secret-sharing schemes. The definitions of access structure and linear secret-sharing schemes are shown as follows.

Access Structure: Let S denote an attribute universe. A collection $A \leq 2^S$ is called monotone if 8B, C € A : if B € A and $B \leq C$, then C € A. A collection (respectively, monotone collection) $A \leq 2^S$ of non-empty subsets of S is an access structure (respectively, monotone access structure) on S. The sets in A are called authorized sets, and the sets not in A are called the unauthorized sets.

Linear Secret-Sharing Schemes (LSSS): Let S be an attribute universe and p be a prime. A secret-sharing scheme π over S is called linear (over Z_p) if (1) The shares of a secret s € Z_p for each attribute form a vector over Z_{pi} (2) For each access structure A on S, there exists a matrix M with l rows and n columns called the share-generating matrix for π. For i = 1, …., l, we define a function labels row i of M with attribute p(i) from S. When we consider the column vector ~v = (s, r_2, ….., r_n), where s € Zp is the secret to be shared and r_2, …., r_n € Z_p are randomly chosen. Then M~v € Z p is the vector of l shares of the secret s according to π. The share (M~v)j "belongs" to attribute (j) for j € [l].

A CP-ABE system consists of four algorithms the following four algorithms:

Setup (λ, U): The setup algorithm takes as input a security parameter and attribute universe U, and outputs a master secret key MSK and the public parameters PP.

Encrypt (PP, A, M): The encryption algorithm takes as input the public parameters PP, an access structure A and a message M, and outputs a ciphertext CT.

KeyGen (MSK, S): The key generation algorithm takes as input the master secret key MSK and an attribute set S, and outputs a secret key SK.

Decrypt (PP, SK, CT): The decryption algorithm takes as input the public parameters PP, a secret key SK and a ciphertext CT. If the attribute set of SK satisfies the access structure of CT, it outputs a message M; otherwise, it outputs?. The definition of CP-ABE's security can be found, which achieves indistinguishability under chosen-plaintext attacks (i.e., is IND-CPA secure).

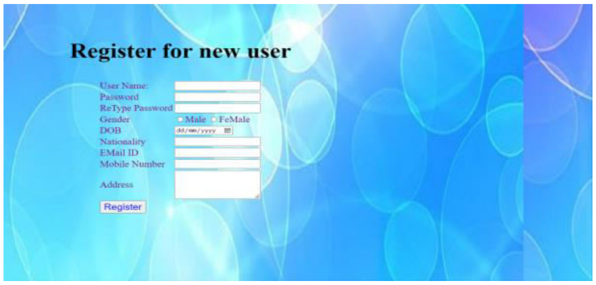
5.1 Authenticated Encryption with Associated Data

Authenticated encryption with associated data (AEAD) is a form of symmetric-key encryption which simultaneously provides confidentiality as well as integrity [28]. A symmetric-key encryption scheme SE mainly consists of the

following two PPT algorithms:

- SE.Enc(m; sk) ! ct: On input a message m and a symmetric key sk, it outputs a ciphertext ct.
- SE.Dec(ct; sk) ! m: On input a symmetric key sk and a ciphertext ct, it outputs a message m.
- An symmetric-key encryption scheme SE should be semantically secure under a chosen plaintext attack.

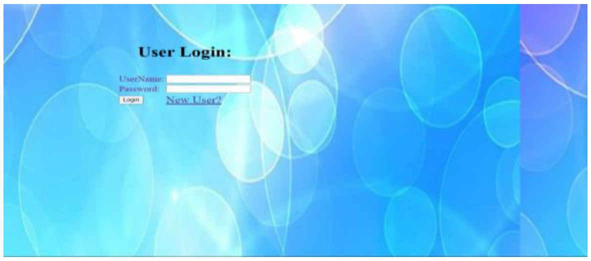
6. RESULT



Register for new user

User Name:
Password
ReType Password
Gender ○ Male ○ FeMale
DOB dd/mm/yyyy
Nationality
EMail ID
Mobile Number
Address

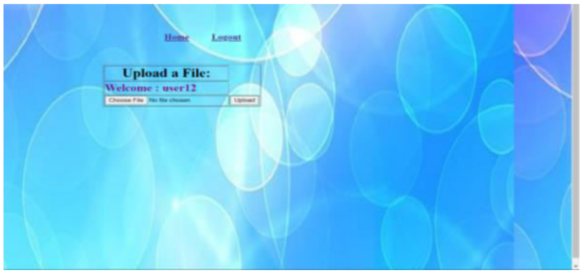
Fig 6. User Registration Page



User Login:

UserName:
Password:

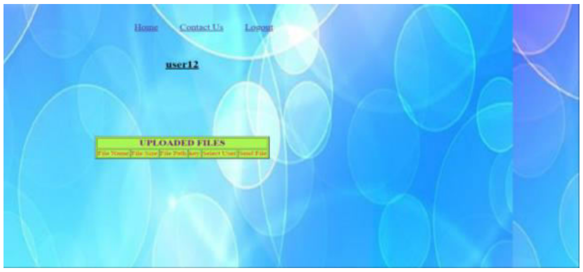
Fig 7. User Login Page



Upload a File:

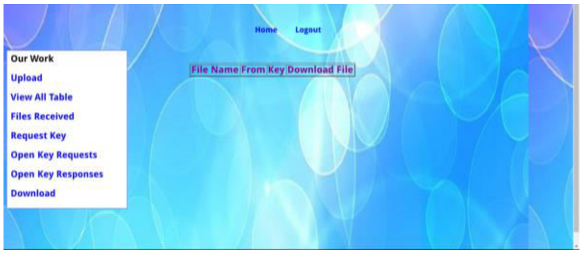
Welcome : user12

Fig 8. Upload Files Page



UPLOADED FILES					
File Name	File Size	File Path	key	Select User	Send File

Fig 9. Storage Status Page



File Name	From Key	Download File

Fig 10. Download Files Page

7. CONCLUSION AND FUTURE ENHANCEMENT

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 request is "transplantable" to other CP-ABE constructions. Our experimental 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 amounts of its 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.

FUTURE WORK: 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 Secure Comm 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] John Bethencourt, Amit Sahai, and Brent 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 provisioning 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 function for cloud storage. IEEE Transactions on Services Computing, 10(5):715–725, 2017.

[16] Jiguo Li, Yao Wang, Yichen Zhang, and Jinguang Han. Full verifiability for outsourced decryption in attribute-based encryption. IEEE Transactions on Services Computing, DOI: 10.1109/TSC.2017.2710190, 2017.

[17] Wei Li, Kaiping Xue, Yingjie Xue, and Jianan Hong. Tmacs: A robust and verifiable threshold multi-authority access control system in public cloud storage. IEEE Transactions on parallel and distributed systems, 27(5):1484–1496, 2016.

[18] Ben Lynn et al. The pairing-based cryptography library. Internet: crypto. stanford. edu/pbc/[Mar. 27, 2013], 2006.

[19] Frank McKeen, Ilya Alexandrovich, Alex Berenzon, Carlos V. Rozas, Hisham Shafi, Vedvyas Shanbhogue, and Uday R. Savagaonkar. Innovative instructions and software model for isolated execution. In HASP@ISCA 2013, page 10, 2013.

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

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

[22] Sankar Das, S. (2024). Harnessing data lineage: making artificial intelligence smarter using data governance Frameworks. International Journal of Research and Analytical Reviews, 11(1). https://doi.org/10.56975/ijrar.v11i1.322571.

[23] Jianting Ning, Zhenfu Cao, Xiaolei Dong, Lifei Wei, and Xiaodong Lin. Large universe ciphertext-policy attribute-based encryption with white-box traceability. In Computer Security-ESORICS 2014, pages 55–72. Springer, 2014.

[24] "Data Science–Enabled Anomaly Detection in Financial Transactions Using Autoencoders and Risk Evaluation Mechanisms," Journal of Information Systems Engineering and Management, 2024, doi: 10.52783/jisem.v9i2.44.

[25] Jianting Ning, Xiaolei Dong, Zhenfu Cao, Lifei Wei, and Xiaodong Lin. White-box traceable ciphertext-policy attribute-based encryption supporting flexible attributes. IEEE Transactions on Information Forensics and Security, 10(6):1274–1288, 2015.

[26] Olga Ohrimenko, Felix Schuster, Cedric Fournet, Aastha Mehta, Sebastian Nowozin, Kapil Vaswani, and Manuel Costa. Oblivious multi-party machine learning on trusted processors. In USENIX Security

[27] Ashay Rane, Calvin Lin, and Mohit Tiwari. Raccoon: Closing digital side-channels through

obfuscated execution. In 24th USENIX Security Symposium, USENIX Security 2015, pages 431–446, 2015.

[28] Phillip Rogaway. Authenticated-encryption with associated-data. In Proceedings of the 9th ACM conference on Computer and communications security, pages 98–107. ACM, 2002.

[29] "Machine Learning–Enhanced Threat Intelligence for Understanding the Underground Cybercrime Market," International Journal of Intelligent Systems and Applications in Engineering, 2022, doi: 10.17762/ijisae.v10i2s.7972.

[30] Ming-Wei Shih, Sangho Lee, Taesoo Kim, and Marcus Peinado. T-sgx: Eradicating controlled-channel attacks against enclave programs. In NDSS 2017, 2017.

[31] Victor Shoup. A proposal for an iso standard for public key encryption (version 2.1). IACR Eprint Archive, 112, 2001. [32] 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.

[33] Mohammed H Sqalli, Fahd Al-Haidari, and Khaled Salah. Edosshield-a two-steps mitigation technique against edos attacks in cloud computing. In UCC 2011, pages 49–56. IEEE, 2011.

[34] Willy Susilo, Peng Jiang, Fuchun Guo, Guomin Yang, Yong Yu, and Yi Mu. Eacsip: Extendable access control system with integrity protection for enhancing collaboration in the cloud. IEEE Transactions on Information Forensics and Security, 12(12):3110–3122, 2017.

[35] Florian Tramer, Fan Zhang, Huang Lin, Jean-Pierre Hubaux, Ari Juels, and Elaine Shi. Sealed-glass proofs: Using transparent enclaves to prove and sell knowledge. In Europe 2017, pages 19–34. IEEE, 2017.

[36] Brent Waters. Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization. In Public Key Cryptography–PKC 2011, pages 53–70. Springer, 2011.

[37] Prodduturi, S.M.K. (2024). 'Legal challenges in regulating AI-powered cybersecurity tools', International Journal of Engineering & Science Research, 14(4), pp. 316–323.

[38] 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.

[39] Shui Yu, Yonghong Tian, Song Guo, and Dapeng Oliver Wu. Can we beat ddos attacks in clouds? IEEE Transactions on Parallel and Distributed Systems, 25(9):2245–2254, 2014.