Two Access Control for Cloud-Based Data Storage and Sharing.

G KEERTESHWAR REDDY, MAHESH PAWAR, D AKHIL
Dept. Computer Science and Engineering
Anurag University
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 mechanisms 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.**

I. INTRODUCTION

Over the past few years, cloud storage services have garnered significant interest from both the academic community and the industrial sector. These services are increasingly being integrated into various online commercial platforms, such as Apple iCloud, because of their numerous advantages, including the ease of access and the elimination of the need for on-premise data management. As a result, a growing number of individuals and businesses are choosing to store their data on remote servers to decrease the expenses associated with the upkeep and enhancement of their own data handling systems. Nonetheless, concerns regarding the security of data stored on these platforms remain a significant barrier to their broader adoption.

In practical scenarios, it often becomes necessary to share stored data with third parties. For instance, a user of Dropbox, referred to here as Alice, might want to share pictures with her friends. To do this without employing encryption, Alice would need to create and distribute a shareable link. While this method does offer a degree of control over who can access the data, thereby preventing unauthorized access (e.g., by people outside Alice's circle of friends), it's possible for the link to be accessed by the cloud service's administrators.

Given the inherent risks associated with storing data on a platform that is inherently insecure due to its connectivity to the internet, experts generally advise encrypting data before uploading it to the cloud. This ensures both the privacy and security of the data. A common approach involves using specific encryption techniques, such as AES, to encrypt the data so that only authorized users with the appropriate decryption key can access it.

When it comes to sharing sensitive materials like photographs, without prior knowledge of who the recipients might be, traditional encryption methods, which require the identity of the decryptor to be known beforehand, fall short. In such instances, a policy-based encryption strategy becomes invaluable. This approach allows the data owner to set access policies directly on the encrypted data, ensuring that only authorized individuals can access it.

Cloud services are also vulnerable to specific attacks, such as resource-exhaustion attacks, which can disrupt service by overwhelming the server with excessive requests. This can lead to an Economic Denial of Sustainability (EDoS) attack, aimed at draining the financial resources of cloud users by exploiting the pay-as-you-go pricing model. Moreover, unlimited access to download requests can potentially expose encrypted data to network attacks, risking unauthorized information disclosure.

To address these issues, this paper introduces a novel concept called dual access control. It incorporates Attribute-Based Encryption (ABE) as a key component, particularly focusing on Ciphertext-Policy ABE (CP-ABE), to secure outsourced data. CP-ABE allows for the establishment of detailed access policies over encrypted data, offering a sophisticated means to control both access and downloading activities. However, merely applying CP-ABE is insufficient for a comprehensive solution that effectively manages both data access and download requests.

II. LITERATURE REVIEW

In recent years, the adoption of cloud-based storage solutions has seen exponential growth due to their significant advantages in cost efficiency and operational

flexibility. This shift has been especially pronounced within both the academic community and various industries, where the demand for scalable, accessible, and cost-effective data storage solutions is ever-increasing. Such platforms, exemplified by services like Apple iCloud, offer compelling benefits, including enhanced accessibility to data and elimination of the need for extensive local data management infrastructure. As a result, a rising number of entities, ranging from individuals to large corporations, are moving towards outsourcing their data storage needs to cloud services. This transition not only facilitates significant cost savings by reducing the need for continuous investment in local data storage and management hardware but also enables users to access their data seamlessly from any location.

Despite these advantages, the migration to cloud-based services is not without its challenges. Chief among these is the concern regarding data security breaches, a critical issue that serves as a considerable deterrent to the wider adoption of cloud storage services. The fundamental issue stems from the inherent risk associated with storing sensitive information on remotely hosted platforms, which, by their very nature, are susceptible to various security threats.

A particular area of concern is the sharing of outsourced data, a common requirement in many practical applications. For instance, a user of a service like Dropbox might wish to share photos with friends. Traditional sharing mechanisms, while providing a level of access control, often do not employ encryption, thereby exposing shared links to potential unauthorized access at the administrative level. To counteract this vulnerability, it's recommended to encrypt data before uploading it to the cloud. This encryption ensures that data privacy and security are maintained, as only users possessing the appropriate decryption keys can access the data. One popular encryption method is the use of Advanced Encryption Standard (AES) for encrypting data before its cloud storage.

However, challenges remain, especially when the need arises to share encrypted data with users whose identities or specific attributes are not known in advance. Traditional public key encryption methods, such as Paillier Encryption, require prior knowledge of the data receiver's identity, making them unsuitable for scenarios where data needs to be shared based on attributes rather than identities. This gap necessitates the adoption of policy-based encryption mechanisms, allowing data owners to define access policies directly on encrypted data. This approach ensures that only authorized users, who meet the defined attributes or policies, can access the encrypted data.

Furthermore, the open nature of cloud services exposes them to various types of attacks, notably resource-exhaustion attacks. These attacks, aimed at overwhelming the cloud server with excessive download requests, can lead to denial-of-service conditions, disrupting access for legitimate users. Such attacks not only impact service availability but also pose significant economic concerns in a pay-as-you-go model, as they can dramatically increase the costs for cloud service users. This phenomenon, known as the Economic Denial of Sustainability (EDoS) attack, highlights the necessity for effective control mechanisms over download requests to prevent unauthorized access and ensure the economic viability of cloud services.

In response to these challenges, the paper proposes a novel dual access control mechanism that leverages the strengths of Ciphertext-Policy Attribute-Based Encryption (CP-ABE). This mechanism offers a robust framework for securing outsourced data in cloud storage, providing both confidentiality and fine-grained control over access to encrypted data. By employing CP-ABE, data owners can define explicit access policies that are enforced directly on the encrypted data, ensuring that only authorized users with matching attributes can access the data. This solution addresses the core security concerns associated with cloud-based storage services, offering a promising approach to enhancing data privacy, security, and overall system resilience against potential attacks.

We positively address the query by introducing two secure and effective systems for dual access control within cloud environments, tailored for various situations. The essence of our approach is to provide an optimized method for dual access control, beginning with the foundation of a Ciphertext-Policy Attribute-Based Encryption (CP-ABE) system, identified as a key component. Additionally, we innovate by implementing a strategic oversight on users' requests to download data, refining the process beyond the conventional "testing" ciphertext approach. Specifically, we enable users to initiate download requests. Upon receipt of such requests, the cloud server, with support from either a designated authority or Intel SGX enclave, verifies the user's eligibility for data access without divulging any extra details aside from the user's authorization status. This underlying framework allows the cloud to effectively manage download requests. The introduced systems are distinguished by several notable attributes:

1. Data Privacy Assurance:Our systems ensure that data encrypted and uploaded to the cloud remains accessible exclusively to those with legitimate authorization.

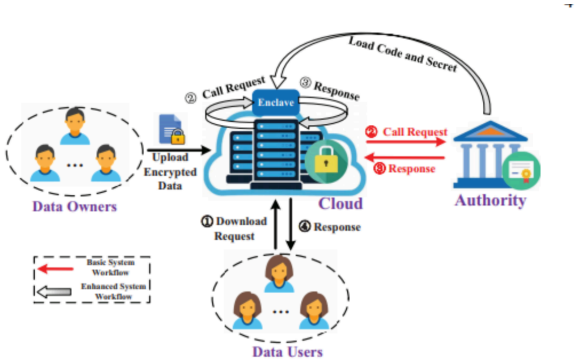
2. Anonymity in Data Sharing: The cloud server is unable to identify the owner of the data being shared, preserving

the data owner's anonymity throughout the storage and sharing process.

3. Granular Access Control: The owner of the data maintains control over their encrypted information by setting specific access policies before cloud upload. This allows only users who meet these criteria to access the encrypted data.

4. Anonymous Download Request Management and Protection Against EDoS Attacks: The cloud server has the capability to manage download requests anonymously, thereby safeguarding the system against Economic Denial of Sustainability (EDoS) attacks.

5. Operational Efficiency: By leveraging the CP-ABE system as a foundation, our proposed solutions do not significantly increase computational or communication burdens, making them practical for real-world deployment when compared with existing systems.



III. PROPOSED METHOD

Notation:

Let PPT be probabilistic polynomial-time. Define $[k] = \{1, 2, ..., k\}$ for $k \in N$. Let $(a1, a2, ..., an)$ be a row vector and $(a1, a2, ..., an)\perp$ be a column vector. By vi we denote the i-th element in a vector $\sim v$. Let $G = (G, GT, p, e)$ be the

groups and the bilinear mapping description, where G and GT are two multiplicative cyclic groups of prime order p and $e : G \times G \rightarrow GT$ is a bilinear map.

Prime Order Bilinear Groups.

Let G and GT be two multiplicative cyclic groups of prime order p. Let g be a generator of G and $e : G \times G \rightarrow GT$ be a bilinear map. The bilinear map e has the following properties: (1) Bilinearity: $\forall u, v \in G$ and $x, y \in Zp$, we have $e(ux, vy) = e(u, v)^{xy}$; (2) Non-degeneracy: $e(g, g)6= 1$. Wesay that G is a bilinear group if the group operations in G and the bilinear map $e : G\times G \rightarrow GT$ can both be computed efficiently.

Complexity Assumption

Assumption 1. *(Decisional q-parallel Bilinear Diffifie-Hellman Exponent assumption (decisional q-Parallel BDHE) [36]) The Decisional q-parallel Bilinear Diffie-Hellman Exponent problem as follows. Initially choose a group G of prime order p according to the security parameter, pick a random group element* $g \in G$, *and* $q + 2$ *random exponents* $c, s, b1, b2, ..., bq \in Zp$. *If an adversary is given the group description* (p, G, GT, e) *and* $\sim z$ *including the following terms:*

$g, gs\ gc, ..., gcq, gcq+2, ..., gc2q\ gs{\cdot}bj, gc/bj, ..., g(cq/bj), g(cq+2/bj), ..., g(c2q/bj)\ \forall 1\leq j\leq q\ gc{\cdot}s{\cdot}bk/bj, ..., g(cq{\cdot}s{\cdot}bk/bj)\ \forall 1\leq j,k\leq q, k\neq j$ *it is hard for the adversary to distinguish* $e(g, g)^{scq+1} \in GT$ *from an element R which is randomly chosen from GT . An algorithm A that outputs* $\beta \in \{0, 1\}$ *has advantage in solving the above assumption if* $|\Pr[A(\sim z, e(g, g)^{scq+1}) = 0] - \Pr[A(\sim z, R) = 0]| \geq$.

Definition 1. *We say that the decisional q-Parallel BDHE assumption holds if no P P T algorithm has a non-negligible*

advantage in solving the decisional q-Parallel BDHE problem.

Ciphertext-Policy Attribute-based-Encryption

Ciphertext-Policy Attribute-based-Encryption (CP-ABE) is a versatile encryption supporting fifine-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 [4], [25]: Let S denote an attribute uni verse. A collection $A \subseteq 2^S$ is called monotone if $\forall B, C \in A$: *if* $B \in A$ and $B \subseteq C$, then $C \in A$. A collection (respectively, monotone collection) $A \subseteq 2^S$ 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) [4], [25]): Let S be an attribute universe and p be a prime. A secret-sharing scheme Q over S is called linear (over Zp) if (1) The shares of a secret $s \in Zp$ for each attribute form a vector over Zp; (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 Q . For $i = 1, ..., l$, we define a function ρ labels row i of M with attribute $\rho(i)$ from S. When we consider the column vector $\sim v = (s, r2, ..., rn)$, where $s \in Zp$ is the secret to be shared and $r2, ..., rn \in Zp$ are randomly chosen. Then $M\sim v \in Z\ l\times 1\ p$ is the vector of l shares of the secret s according to Q . The share $(M\sim v)j$ "belongs" to the attribute $\rho(j)$ for $j \in [l]$.

A CP-ABE system consists of four algorithms the follow

ing four algorithms:

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

• **Encrypt(***P P,* A*, M***).** The encryption algorithm takes as input the public parameters *P P*, 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(***P P, SK, CT***).** The decryption algorithm takes as input the public parameters *P P*, 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 $\perp$.

The definition of CP-ABE's security can be found in [36], which achieves indistinguishability under chosen-plaintext attacks (i.e., is IND-CPA secure).

Authenticated Encryption with Associated Data

Authenticated encryption with associated data (AEAD) is a form of symmetric-key encryption which simultaneous ly 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 se semantically secure under a chosen plaintext attack.

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. *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.

We utilize a hybrid security system that integrates the speed of symmetric-key encryption with the flexibility of public-key encryption. This approach involves two layers of access control in a Key/Data Encapsulation Mechanism (KEM/DEM) framework. The bulk of the message is encrypted using a fast symmetric-key algorithm, while a slower public-key method, specifically CP-ABE, is employed solely for encrypting and decrypting a small key.

For ensuring the privacy of shared data, maintaining its confidentiality, and controlling access, we rely on the CP-ABE method as the core component. We adopt a CP-ABE scheme known for its efficiency and sophisticated design to meet these security needs. Moreover, to cater to the anonymous requests for data downloads and manage access without compromising sensitive details (such as the user's identity or the data's unencrypted form), we have devised a system where the cloud can verify a user's authorization status with no need to access private information. Initially, this verification process requires assistance from an authority, necessitating its constant availability. Nonetheless, we acknowledge that this may not be feasible in all real-world scenarios, leading us to consider alternative approaches in certain cases.

In scenarios where maintaining constant online presence of the authority is not feasible, we introduce an improved model. This model permits the authority to go offline after setting up the necessary parameters. To facilitate this, we incorporate the SGX (Software Guard Extensions) technology, which effectively takes over the authority's role in managing access control during download requests.

The foundation of our systems is aimed at ensuring robust security and privacy for data shared on the cloud, protecting against threats such as EDoS attacks. Our strategy involves implementing a dual access control mechanism, as outlined earlier. We build upon the CP-ABE framework suggested in prior studies, adjusting it for use in a KEM/DEM configuration. However, integrating CP-ABE directly into KEM/DEM does not fully address the need for dual access control. Therefore, we introduce an innovative method that eliminates the

need for the "testing" ciphertext used in preliminary models. This method involves the data owner creating a download request that incorporates a randomized version of their secret key. This key retains its ability to decrypt, allowing the cloud to verify if the requestor has the authority to access the encrypted data without revealing the data owner's identity. Hence, the download request serves as an anonymous means to ascertain the legitimacy of the data owner's access rights without compromising their anonymity.

To ensure the cloud does not access sensitive information during this verification process, either an authority's intervention or the use of an Intel SGX enclave is necessary. Our first system relies on the authority's participation for this verification, whereas our second system utilizes Intel SGX for a more autonomous approach. This technique is versatile and can be adapted to most existing CP-ABE frameworks that employ bilinear maps.

The architectures of our dual access control systems for Cloud data sharing is shown in Fig. 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).

The description of workflow is introduced as follows. 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 ① in Fig.), the cloud does as follows.

(a) For our basic system, the cloud sends a call request to the authority (see red ② between the cloud and the authority in Fig.). After receiving a response from the authority (see red ③ between the cloud and the authority in Fig.), the cloud sends a response back to the data user (see ④ in Fig.).

(b) For our enhanced system, the cloud sends a call request to the enclave (see black ② above the cloud in Fig.). After receiving a response from the enclave (see black ③ above the cloud in Fig.), the cloud sends a response back to the data user (see ④ in Fig.).

In the basic construction, the authority must be always online. It is desirable that the cloud can check the download request by itself. In this subsection, to

To address this issue, we present an enhanced system. The procedures Data User Registration, Shared File Generation and Outsourcing, Download Request Generation, Access Shared Data are the same as those of the basic system, the remaining algorithms are modified as follows.

• Parameter Initialization: This procedure is almost the same with that of the basic system, excepting for the following additional steps (that follows the last step of the basic system): – The cloud equipped with SGX processors creates an enclave 4. – The authority prepares a SGX program C for realizing the following functionality: Upon receiving an input h, compute E01 = (h)s0 and output E01 , where s0 is the internal secret inside an enclave. – The authority establishes a secure channel with the enclave, and securely loads the code of program C and the master secret parameter a to the enclave, using for instance AES-GCM for confidentiality and integrity protection [26] (In particular, the authority uses a randomly generated secret key to encrypt the code and the data, and employs the secure channel to share the secret key with the enclave). – The enclave keeps a as its internal secret (i.e., sets a = s0).

In order to verify the software running in the enclave on the cloud side, the authority uses remote attestation [2] to check the integrity of the code (i.e., the program C) and static data (i.e., the master secret parameter a) loaded into the enclave [26] (please refer to Subsection 2.6 for more details about remote attestation).

• Access Control on Download Request :

The procedure is almost the same as that of the basic system, excepting for replacing the first step with the following steps:

– The cloud sends a call request to the enclave with C2 (of CT) as input.

– Upon receiving the call request with C2, the enclave runs program C with C2 as input (i.e., calculates E01 = (C2)a) and returns E01 to the cloud. – The cloud computes E1 = e(E01 , L02) = e(g, g)^savr.

Side-channel resilience. Although the security of SGX is evolving, it is still susceptible to a number of side-channel attacks [6], [14], [30], [37]. One defense against these side channel attacks is to ensure that the enclave program is data-oblivious. That is, the program will not include control flow branches or memory access patterns that depend on the values of sensitive data [7], [13]. Another approach is to employ the technique of ORAM [27]. For the enhanced system, the only enclave operations that touch secret data are decryption operations (for loading the data via AESGCM) and the specific function (that compute

E01 = (h)s0 and output E01). In our implementation of AES-GCM, we utilize the SGX SDK cryptographic library, therefore, it is resilient to software-based side-channels (which is similar to [7]). For the function, we implemented it in a way that it achieves the property of data-oblivious (i.e., control flow branches or memory access patterns will not depend on the sensitive data). Therefore, the enhanced system is secure against side channel attack.

IV. Results And Discussion

Experimental Analysis To evaluate the practical performance, we implement the two proposed systems within the Charm framework [1], where 224-bit MNT elliptic curves from Pairing-Based Cryptography library [18] are used. The experiments are performed in test beds of two PCs. The first PC plays the roles of data owner and data user, the second PC plays the role of authority and cloud. The hardware and software of the first PC are as follows: Intel Core i7-7700M CPU @3.6 GHz,Since the two proposed systems are built on the top of the CP-ABE system in [36], in this subsection, we first give a theoretical analysis of the comparison between the two proposed systems and the (underlying) CP-ABE system in [36]. Let Σ0, Σ1, Σ2 be the CP-ABE system in [36], the basic system in subsection 4.2 and the enhanced system in subsection 4.3, respectively. Table 1 gives the comparison in terms of computational cost. In particular, the computational cost of Parameter Initialization of Σ1 (resp. Σ2) is the same (resp. almost the same) as the algorithm **Setup**(λ, U) of Σ0, excepting that it adds a into the master secret key MSK. Furthermore, the generation of secret keys of Σ1 (resp. Σ2) is the same as that of Σ0. In addition, the computational costs of encryption and decryption of Σ1 (resp. Σ2) are the same as that of Σ0 5 . That is, compared with Σ0, the two proposed systems do not impose any additional computational cost. Table 2 gives the comparison in terms of communication cost. In particular, the public parameters size, secret key size, ciphertext size of Σ1 (resp. Σ2) are all the same with that of Σ0. We note that the technique used to fulfill the feature of access control on download requests is “transplantable” to other CP-ABE. Table 3 gives the comparison among the strawman approach described in the Introduction, our proposed systems and the related work in terms of computational cost. For a fair comparison, for each computational cost in [38], we only count the computational cost which is used for access control on download requests. The computational costs for procedures Parameter Initialization, Shared File Generation and Outsourcing and access control on download request on the data user side of our proposed systems are less (or much less) than that of [38]. In contrast, the access control on download requests on the cloudside of our proposed systems require more computations.This exactly reflects the main design philosophy: to move expensive computations to the cloud as much as possible. comparison among the strawman approach described in the Introduction, our proposed systems and the related work in terms of communication cost. For a fair comparison, we only count the communication cost in [38] which is used for access control on download requests. It shows that the communication cost for download requests of our proposed systems are less than that of [38]. In particular, the ciphertext size of our proposed systems are much less.

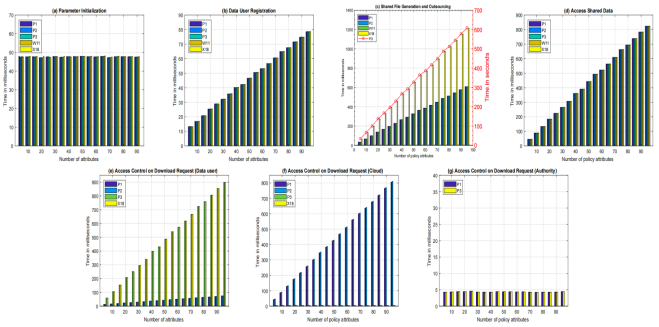


Fig. 2 Experimental results in terms of computational cost

V. Conclusion

We tackled a persistent challenge in the realm of cloud data sharing by developing two dual access control systems designed to withstand DDoS/EDoS attacks. We highlight that the mechanism enabling control over download requests can be adapted and applied to other CP-ABE frameworks. Our tests confirm that these systems introduce minimal additional computational and communication burdens in comparison to the foundational CP-ABE elements they build upon.

In our advanced system, we leverage the security feature that prevents the extraction of confidential information once it's within the enclave. However, recent studies indicate the possibility of enclaves revealing some secrets to a hostile host via memory access patterns or through other forms of side-channel attacks. This has led to the proposal of a model that ensures transparent execution within enclaves. Developing a dual access control system that operates on the principles of transparent enclave execution for cloud data sharing poses an intriguing challenge. Addressing this in our forthcoming research is a priority.

V. 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] 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.