iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 11 **Issue:** IV **Month of publication:** April 2023

DOI: https://doi.org/10.22214/ijraset.2023.50403

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 11 Issue IV Apr 2023- Available at www.ijraset.com

A Dual Security Protection Mechanism for Cloud-Based Data Storage and Sharing

Prof. Kaveri Deosarkar[1], Mr. Kamlesh Samrit[2]
[1]*Asst. Professor, Computer Science Engineering, WCEM, Nagpur, India*
[2]*M.Tech Second Year, WCEM, Nagpur, India*

Abstract: ***Cloud-based data storage service has drawn increasing interest from both academic and industry in recent years due to their 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.***
The most widely used method is encryption to protect sensitive data from being compromised. However, simply encrypting data (e.g., via AES) cannot fully address the practical need for data management. Besides, effective access control over download requests also needs to be considered so that Economic Denial of Sustainability (EDoS) attacks cannot be launched to hinder users from enjoying service.
In this project, we consider 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 requests without loss of security and efficiency. Two dual access control systems are designed in this paper, each for a distinctly designed setting. The security and experimental analysis for the systems are also presented.
Keywords: security, AES, EDoS, Cloud-based data sharing, access control, cloud storage service, Intel SGX, attribute-based encryption

I. INTRODUCTION

In 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. The increasing number of individuals and companies nowadays prefer to outsource their data to the 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 breaches over outsourced data may be one of the main obstacles hindering Internet users from widely using cloud-based storage services.

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 who are not Alice's friends), the sharing link may be visible within the Dropbox administration level (e.g., an administrator could reach the link). Since the cloud (which is deployed in an open network) is not 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 the cloud, so that only a specified cloud user (with a valid decryption key) can gain access to the data via valid decryption.

To prevent shared photos from being accessed by the "insiders" of the system, a straightforward way is to designate the group of authorised data users before encrypting the data. In some cases, Alice may have no idea about whom the photo receivers/users will 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 a policy-based encryption mechanism over the outsourced photos is therefore desirable, so Alice uses the mechanism to define an access policy over the encrypted photos to guarantee only a group of authorised users can access the photos.

In a cloud-based storage service, a common attack is well-known as a resource-exhaustion attack. Since a (public) cloud may not have any control over download requests (namely, a service user may send unlimited numbers of download requests to a 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 the Economic Denial of Sustainability (EDoS) attack [32], [33], which targets the cloud adopter's economic resources.

©IJRASET: All Rights are Reserved

Apart from economic loss, the 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, effective control over download requests 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 the 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 requests is to leverage dummy ciphertexts to verify the data receiver's decryption rights. It, concretely, requires the data owner, say, Alice, to upload multiple "testing" ciphertexts along with the "real" encryption of data to the cloud, where the "testing" ciphertexts are the encryptions of dummy messages under the same access policy as that of the "real" data. After re-receiving a download request from a user, say, Bob, the 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 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 exam- ple, Alice just wants to upload one photo to iCloud from her cellphone but needs to prepare more than one cypher-text. Second, all ciphertexts, including dummy ones, are uploaded to the cloud at the same time. This inevitably imposes an 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 a pay-as-you-go plan or equipped with the 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 the cloud, as a test of his valid download request. As a result, Bob has to "pay" double (decryption price) for access to the "real" data, which again may not be scalable in a resource-constrained setting. Therefore, this paper raises the following question:

"Does there exist a cloud-based mechanism supporting dual access control (over both fine-grained data access and download request) without loss of security and efficiency?"

II. RELATED WORK

To apply fine-grained policy-based control over encrypted data, ABE [9], [29] has been introduced in the literature. Concretely, ABE has two main research branches: one is CP-ABE, and the other is KP-ABE which refers to key- policy ABE. This paper mainly deals with the former. In a CP-ABE, a decryption key is associated with an attribute set and ciphertext is embedded with the access policy. This feature makes CP-ABE quite suitable for secure cloud data sharing (compared to KP-ABE). Note this is so because KP-ABE requires a decryption key to be associated with the access policy which yields heavy storage costs for cloud users. Since the introduction of seminal CP-ABE [9], many works have been proposed to employ CP-ABE in various applications, e.g., accountable, and traceable CP-ABE [22], [23], [24], [25], multi-authority [10], [17], outsourced CP-ABE [15], [16], [21], and extendable variants [34].

Although being able to support fine-grained data access, CP-ABE, acting as a single solution, is far from practical and effective to hold against EDoS attacks [11] which is the case of DDoS in the cloud setting [11], [39]. Several counter-measures to the attack [12], [33] have been proposed in the literature. But Xue et al. [38] stated that the previous works could not fully defend against the EDoS attack at the algorithmic (or protocol) level, and they further proposed a solution to secure cloud data sharing from the attack. However, [38] suffers from two disadvantages. First, the data owner is required to generate a set of challenge ciphertexts in order to resist the attack, which enhances its computational burden. Second, a data user is required to decrypt one of the challenge ciphertexts as a test, which costs plenty of expensive operations (e.g., pairing). Here the computational complexity of both parties is inevitably increased and meanwhile, high network bandwidth is required for the delivery of ciphertexts. The considerable computational power of the cloud is not fully considered in [38]. In this paper, we will present a new solution that requires less computation and communication costs to stand till in front of the EdoS attack. Recently, Antonis Michalas [20] proposed a data-sharing protocol that combines symmetric searchable encryption and ABE, which allows users to directly search over encrypted data. To implement the functionality of key revocation in ABE, the protocol utilizes SGX to host a revocation authority.

Bakas and Michalas [3] later extended the protocol [20] and proposed a hybrid encryption scheme that reduces the problem of multi-user data sharing to that of a single user. In particular, the symmetric key used for data encryption is stored in an SGX enclave, which is encrypted with an ABE scheme. Similar to [20], it deals with the revocation problem in the context of ABE by employing the SGX enclave. In this work, we employ SGX to enable the control of the download request (such that the DDoS/EDoS attacks can be prevented). In this sense, our purpose and our technique of ours are different from that of the protocols in [3], [20].

III. PROBLEM STATEMENT

Nowadays Cloud storage is understood as a promising resolution for providing convenient, universal, and on-demand access to bigger amounts of knowledge shared on the net. In the existing system, they introduced a two-factor security protection mechanism for information to keep within the cloud. The system is predicated on Identity-Based cryptography (IBE) mechanism. The sender needs solely the identity of the receiver to send associate encrypted information. The sender sends cypher text through the cloud to the receiver then the receiver will transfer the cypher text at any time. The existing system accommodates a two-factor encryption protection technique. Encrypted information keep in the cloud, receiver accessed encrypted information and convert it into decrypted information which points to it'll need 2 things: very first thing, the user's secret key that is sent by the sender through a secure channel (e.g., email). The second issue, a user desires a distinctive personal security device to attach to the pc like USB. The system user needed a security device then it'll request the security device to the protection device establishment (SDI) suppose the device is stolen or loss then the user report back to SDI, subsequently establishing revoked personal security device of the user and affording a brand new distinctive or personal security device to the user.

IV. THE PROPOSED SYSTEMS

A. *System Overview*

We employ the use of a hybrid system to protect the data, which combines the efficiency of a symmetric-key system with the convenience of a public-key system. In particular, the proposed dual access control systems are both in Key/Data Encapsulation Mechanism (KEM/DEM) setting [31]. The message is encrypted by an efficient symmetric-key encryption scheme, while the inefficient public-key scheme (i.e., the CP-ABE) is used only to encrypt/decrypt a short key value.

To achieve the security requirements of anonymous data sharing, confidentiality of shared data and access control on shared data, we employ the CP-ABE technique as the basic building block. Specifically, we present the construction based on the CP-ABE scheme in [36] due to its efficiency and elegant construction. To achieve the security requirements of anonymous download request and access control on download request, we design an effective mechanism that the cloud can judge whether a data user is authorized or not without revealing any sensitive information (including the identity of the data user, the plaintext of the outsourced data) to it. In the first system, the cloud needs the help of the authority during the judgement on the download request (sent by a data user). As a result, the authority needs to be always *online*. However, in some other cases in practice, the authority may not be always online. This leads to the second (enhanced) system where the authority can be offline after the parameter initialization procedure. In particular, we employ the SGX technique to replace the role of the authority during the access control on download request procedure.

We now explain the rationale behind our proposed systems. In order to provide strong security and privacy guarantees for shared data on the cloud (that could defend the EDoS attack), a cloud-based data sharing system should support dual access control as described in Section 1. We start from the CP-ABE system proposed in [36], and adapt it to the KEM/DEM setting. However, simply employing the CP-ABE construction from [36] in the KEM/DEM setting is not sufficient to provide dual access control. New technique needs to be introduced such that the control of both data access and download request can be guaranteed. Different from the strawman solution described in Section 1, we introduce a new approach to avoid using the “testing” ciphertext in the strawman solution. Specifically, we allow the data owner to generate a download request, which contains a randomized form of the secret key held by the data owner. The download request retains the “decryption capability” of the secret key such that it can be used to test whether the underlying data owner is capable to decrypt the shared ciphertext(s). Since the above-mentioned component contained in the download request is randomized, it cannot be utilized to infer the owner of the secret key. That is, the download request enables the cloud to check whether the data owner of the download request is authorized without leaking the identity of the underlying data owner (i.e., the download request is anonymous). To further prevent leaking secret information to the cloud, the verification of download requests needs the help of the authority or the enclave of Intel SGX.

Our first system is designed for the case where the verification of a download request involves the help of the authority, while the second system is designed for the case where the enclave of Intel SGX is involved during the verification of the download request procedure. We note that our technique described above is general in the sense that it can be applied to most of the current CP-ABE constructions based on bilinear maps.

B. The enhanced System

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

1) 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

- The authority prepares a SGX program C for realizing the following functionality: Upon receiving an input h, compute E1′ = (h)s′ and output E1′ , where s′ 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 an as its internal secret (i.e., sets a = s′).

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).

2) 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 C T) as input.
- Upon receiving the call request with C2, the enclave runs program C with C2 as input (i.e., calculates E1′ = (C2)a) and returns E1′ to the cloud.
- The cloud computes E1 = e(E1′ , L′2) = e(g, g).

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 defence 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 AES- GCM) and the specific function (that computes E1′ = (h)s′ and output E1′). 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 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 attacks.

V. SECURITY ANALYSIS

In this section, we present the security analyses of the two proposed systems on how they achieve the security requirements.

A. Security of the basic system

Security against honest-but-curious cloud

For simplicity, we denote by Γ1, Γ2 the CP-ABE scheme in [36] and the basic system in Subsection 4.2, respectively.

Lemma 1. *[36] If the decisional* q*-Parallel BDHE assumption holds,* Γ1 *is IND-CPA secure.*

Lemma 2. *If* Γ1 *is IND-CPA secure,* Γ2 *is IND-CPA secure.*

Proof. To prove the security of Γ2, we suppose there exists a PPT adversary A2 with a challenge access policy $(M*, \rho*)$ ($M*$ is an $l \times n$ matrix) that has a non-negligible advantage in breaking Γ2. We build a PPT simulator algorithm A1 that has a non-negligible advantage in breaking Γ1.

Init: A1 gets the challenge access policy $(M*,\rho*)$ from A2 and sends the received $(M*, \rho*)$ to the Γ1 challenger.

Setup: A1 receives the public parameters pk = (g, ga, h1, ..., hU , e(g, g)α) from the Γ1 challenger. It sends pk to A2.

Phase 1: For the secret key query from A2 with an attribute set S (with a restriction that S does not satisfy $(M*, \rho*)$), A1 sends it to the Γ1 challenger and obtains a secret key SKS′ = (K′,L′,{Kx′ }x∈S). A1 then sets L1 = K′,L2 = L′,{L3,x = Kx′ }x∈S) and returns SK = (L1,L2,{L3,x}x∈S)) to A2.

Challenge: A2 declares two equal-length messages (m0,m1) and sends them to A1. A1 chooses two random symmetric key S K0 , S K1 , sends them to the Γ1 challenger and obtains a challenge ciphertext $CT* = (C*, C'*, \{Cx*,1 , Cx*,2 \}x\in[l])$. A1 selects a random bit $bA1 \in \{0, 1\}$, and runs SE.Enc(M, SKbA1) to obtain.

Guess: A2 outputs a guess $b \in \{0, 1\}$ and sends it to A1. A1 sends the received b to the Γ1 challenger.

Note that the distributions of the public parameters, chal- lenge ciphertext and decryption keys in the above game are the same as that of the real system, if A2 can break Γ2 with a non-negligible advantage, A1 can break Γ1 with the same advantage.

Lemma 3. *If the decisional* q*-Parallel BDHE assumption holds, the cloud cannot identify the owner of any newly uploaded file.*

Proof. From Lemma 1 and Lemma 2, we have that Γ2 is IND-CPA secure. Hence, the shared file(CT,CT) does not contain any information that can be used to make inferences about the owner of the file. We conclude that the cloud cannot obtain any useful information to know the owner.

Lemma 4. *If the decisional* q*-Parallel BDHE assumption holds, the cloud cannot obtain the plaintext of the encrypted data stored on it.*

Proof. According to the *Shared File Generation and Outsourcing* phase, we know that the encrypted data (CT,CT) is generated based on a hybrid system of the symmetric-key encryption scheme SE and the CP-ABE scheme Γ1 [36]. From Lemma 1 and Lemma 2, we have that the basic system in Subsection 4.2 Γ2 is IND-CPA secure. Due to the security property of Γ2, it follows from the security result of hybrid encryption system [8] that the encrypted data (C T , C T) can only be decrypted with valid secret keys. Since the cloud cannot obtain such secret keys, it cannot decrypt the file.

Lemma 5. *If the decisional* q*-Parallel BDHE assumption holds, the cloud cannot identify the sender of any download request.*

Proof. From Lemma 1 and Lemma 2, we have that the basic system in Subsection 4.2 is IND-CPA secure. Hence, for any download request DReq = (′download′, (L′1,L′2,{L′3,x}∀x∈S,S)), it does not leak any information that can be used to make inference about its sender. We con- clude that the cloud cannot obtain any useful information to identify the sender.

Theorem 1. *The basic system in Subsection 4.2 is secure against honest-but-curious cloud.*

Proof. It follows directly from Lemma 3, Lemma 4, and Lemma 5.

B. Security against malicious data user

Lemma 6. *If the decisional* q*-Parallel BDHE assumption holds, any unauthorized data user cannot download the shared file(s).*

Proof. In order to download a shared file (CT ,CT) (where CT = (C1, C2, {D1,i, D2,i}i∈[l], (M, ρ))) from the cloud, a download request sent by any data user has to pass the check on the cloud side. Specifically, for a download request DReq = (′download′, (L′1,L′2,{L′3,x}∀x∈S,S)), it passes the check on the cloud side if the following two conditions are satisfied: (1) (M, ρ) is satisfied by S; and (2) the equation E1 = E2 holds, where E1 = e((C2)a, L′2), E2 = □i∈I (e(D1,i , L′2)e(D2,i , L′3,ρ(i)))wi (as described in Section 4). Since the authority is fully trusted, we have that E1 = e((C2)a, L′2) = e(g, g)savr. Suppose there exists an adversary (i.e., unauthorized data user) that can con- struct a download request such that the equation E2 = e(g,g)avsr (i.e., E2 = E1) holds during the procedure Access Control on Download Request. That is, the adversary can construct a download request that satisfies the above conditions (1) and (2). It implies that the adversary can construct a download request that is derived from a valid secret key. It implies that the adversary can construct such valid secret key, which breaks the IND-CPA security of the basic system in Subsection 4.2. However, from Lemma 1 and Lemma 2, we have that the basic system in Subsection 4.2 is IND-CPA secure.

Lemma 7. *If the decisional* q*-Parallel BDHE assumption holds, any unauthorized data user cannot decrypt the shared file even if the data user obtains the file.*

Proof. It follows directly from Lemma 2.

Theorem 2. *The basic system in Subsection 4.2 is secure against malicious data user.*

Proof. It follows directly from Lemma 6 and Lemma 7. **5.2 Security of the enhanced system**

Security against honest-but-curious cloud

Theorem 3. *The enhanced system in Subsection 4.2 is secure against honest-but-curious cloud.*

Proof. The proof of this theorem is the same with that of Theorem 1.

Security against *malicious* data user

Let Γ′1, Γ′2 be the CP-ABE scheme in [36] and the enhanced system in Subsection 4.3, respectively.

Lemma 8. *If* Γ′1 *is IND-CPA secure,* Γ′2 *is IND-CPA secure.*

Proof. Since the procedures Data User Registration, Shared File Generation and Outsourcing of Γ′2 are the same as those of the basic system, the proof of this lemma is the same with that of Lemma 2.

Lemma 9. *Any unauthorized data user cannot download the shared file(s).*

Proof. Similar to the basic system, in order to download a shared file (CT , CT) (where CT = (C1, C2, {D1,i, D2,i}i∈[l], (M, ρ))) from the cloud, a download request sent by any data user has to pass the check on the cloud side. Specifically, for a download request DReq = (′download′, (L′1,L′2,{L′3,x}∀x∈S,S)), it can pass the check on the cloud side if the following two conditions are satisfied: (1) (M,ρ) is satisfied by S; and (2) the equation E1 = E2 holds, where E1 = e((C2)a,L′2), E2 = □i∈I (e(D1,i , L′2)e(D2,i , L′3,ρ(i)))wi (as described in Section 4). Due to the isolation functionality of SGX, the code and data (i.e., C and a) inside the enclave-protected memory cannot be modified by any process external to the enclave. Hence, E1′ always equals (C2)a and E1 always equals e(g, g)svr. Suppose there exists an unauthorized adversary that can construct a download request such that the equation E2 = e(g,g)avsr (i.e., E2 = E1) holds during the procedure Access Control on Download Request, i.e., the above conditions (1) and (2) are satisfied. It implies that the adversary can construct a download request that is derived from a valid secret key. Since the adversary is unauthorized, it implies that the adversary can construct such a valid secret key, which breaks the IND-CPA security of the enhanced system in Subsection 4.3. However, from Lemma 1 and Lemma 8, we have that the enhanced system in Subsection 4.3 is IND-CPA secure.

Lemma 10. *Any unauthorized data user cannot decrypt the shared file even if the data user obtains the file.*

Proof. It follows directly from Lemma 8.

Theorem 4. *The enhanced system in Subsection 4.3 is secure against malicious data users.*

Proof. It follows directly from Lemma 9 and Lemma 10.

VI. CONCLUSION AND FUTURE WORK

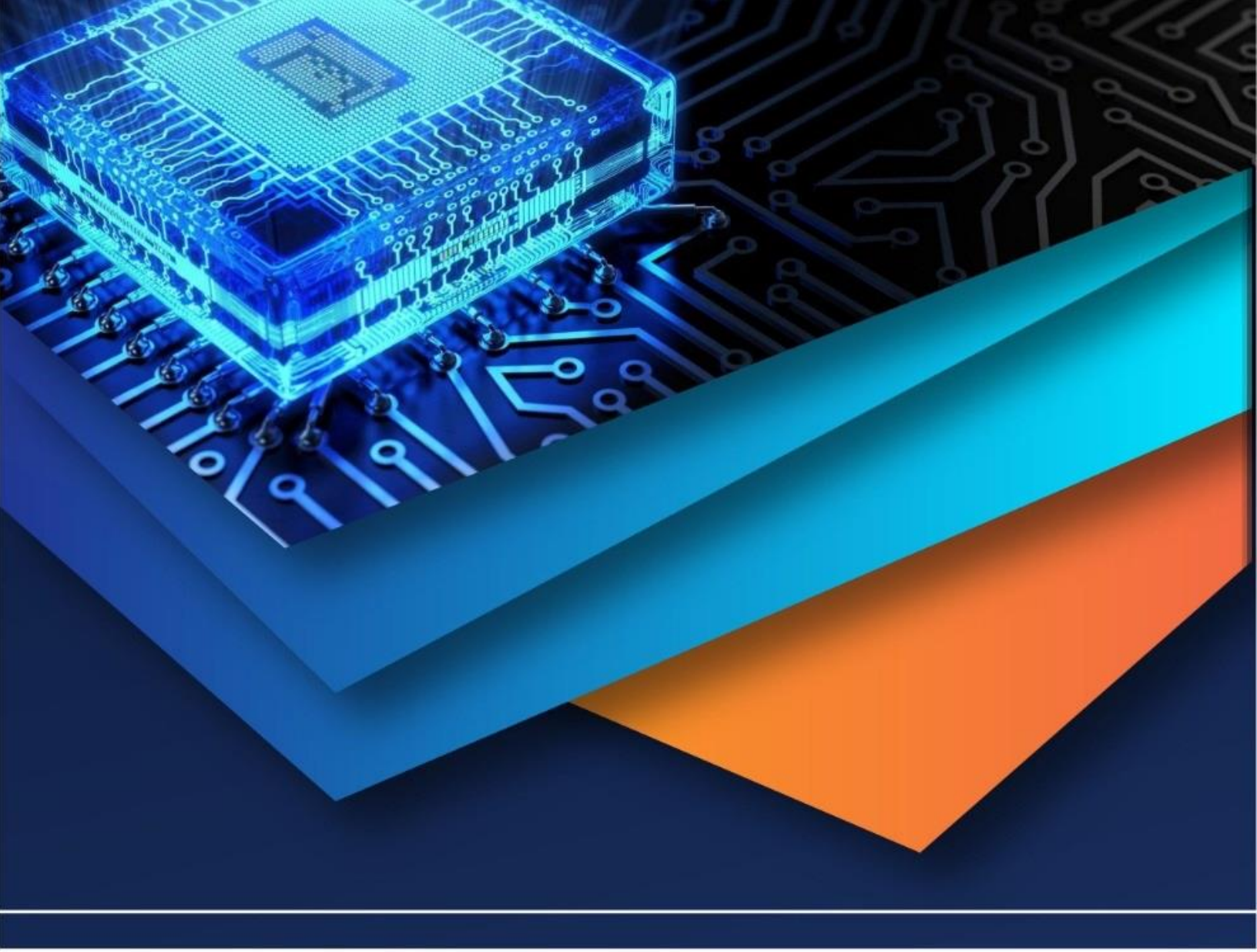
We addressed an interesting and long-lasting problem in cloud-based data sharing and presented two dual access control systems. The proposed systems are resistant to DDoS/EDoS attacks. We state that the technique used to achieve the feature of control on download requests is “transplantable” to other 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 enclaves may leak some amounts of its secret(s) to a malicious host through memory access patterns [37] or other related side-channel attacks [14], [30]. The model of *transparent enclave execution* is hence introduced in [35]. Constructing a dual access control system for cloud data sharing from a 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. Jour- nal 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 priva- cy (HASP), volume 13, page 7. ACM New York, NY, USA, 2013.
[3] Alexandros Bakas and Antonis Michalas. Modern Family: A revo- cable hybrid encryption scheme based on attribute-based encryption, symmetric searchable encryption and SGX. In SecureComm 2019, pages 472–486, 201
[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 en- crypted 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 (distribut- ed 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. Ksf- oabe: 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. Ful- l 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. Sava- gaonkar. 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 shar- ing. In SAC 2019, pages 146–155, 2019.
[21] Jianting Ning, Zhenfu Cao, Xiaolei Dong, Kaitai Liang, Hui Ma, and Lifei Wei. Auditable σ-time outsourced attribute-based en- cryption for access control in cloud computing. IEEE Transactions on Information Forensics and Security, 13(1):94–105, 2018.
[22] JiantingNing,ZhenfuCao,XiaoleiDong,andLifeiWei.White-box traceable CP-ABE for cloud storage service: How to catch people leaking their access credentials effectively. IEEE Transactions on Dependable and Secure Computing, 15(5):883–897, 2018.
[23] JiantingNing,ZhenfuCao,XiaoleiDong,LifeiWei,andXiaodong Lin. Large universe ciphertext-policy attribute-based encryption with white-box traceability. In Computer Security-ESORICS 2014, pages 55–72. Springer, 2014.
[24] Jianting Ning, Xiaolei Dong, Zhenfu Cao, and Lifei Wei. Ac- countable authority ciphertext-policy attribute-based encryption with white-box traceability and public auditing in the cloud. In Computer Security–ESORICS 2015, pages 270–289. Springer, 2015.
[25] Jianting Ning, Xiaolei Dong, Zhenfu Cao, Lifei Wei, and Xiaodong Lin. White-box traceable ciphertext-policy attribute-based encryp- tion supporting flexible attributes. IEEE Transactions on Information Forensics and Security, 10(6):1274–1288, 2015.
[26] Olga Ohrimenko, Felix Schuster, Ce´dric Fournet, Aastha Mehta, Sebastian Nowozin, Kapil Vaswani, and Manuel Costa. Oblivious multi-party machine learning on trusted processors. In USENIX Security Symposium, pages 619–636, 2016. 1545-5971 (c) 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information. Authorized licensed use limited to: UNIVERSITY OF BIRMINGHAM. Downloaded on July 26,2020 at 07:27:11 UTC from IEEE Xplore. Restrictions apply.
[27] Ashay Rane, Calvin Lin, and Mohit Tiwari. Raccoon: Closing dig- ital 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 communi- cations security, pages 98–107. ACM, 2002.
[29] Amit Sahai and Brent Waters. Fuzzy identity-based encryp- tion. In Advances in Cryptology–EUROCRYPT 2005, pages 457–473. Springer, 2005.
[30] Ming-Wei Shih, Sangho Lee, Taesoo Kim, and Marcus Peinado. T-sgx: Eradicating controlled-channel attacks against enclave pro- grams. 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. D- dos/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. Edos- shield-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 in- tegrity 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 EuroS&P 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] Yuanzhong Xu, Weidong Cui, and Marcus Peinado. Controlled- channel attacks: Deterministic side channels for untrusted operat- ing systems. In S&P 2015, pages 640–656. IEEE, 2015.
[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.



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)