EasyChair Preprint

№ 12793

Dual Access Control for Cloud-Based Data Storage and Sharing

Varun Jammula and Nidhi Shah

March 27, 2024

Dual Access Control for Cloud-Based Data Storage and Sharing

JAMMULA VARUN
Computer Science Engineering (CSE),
Parul University, Gujarat, India
jammulavarun143@gmail.com

NIDHI SHAH,Professor
Dept. Of Computer Science Engineering
Parul University, Gujarat, India
nidhi.shah19176@paruluniversity.ac.in

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.**

I. INTRODUCTION

In the recent decades, cloud-based storage service hasattracted 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 toremote cloud in such a way that they may reduce the cost of upgrading their local data management facilities/devices.However, the worry of security breach over outsourced data may be one of the main obstacles hindering Internet users from widely using cloud-based storage service.

In many practical applications, outsourced data may need to be further shared with others. For example, a Dropbox user Alice may share photos with her friends.Without using data encryption, prior to sharing the photos,Alice needs to generate a sharing link and further share the link with friends. Although guaranteeing some level of access control over unauthorized users (e.g., those are not Alice's friends), the sharing link may be visible within the Dropbox administration level (e.g., administrator could reach the link).

Since the cloud (which is deployed in an open network) is not be fully trusted, it is generally recommended to encrypt the data prior to being uploaded to the cloud to ensure data security and privacy. One of the corresponding solutions is to directly employ an encryption technique (e.g.,AES) on the outsourced data before uploading to cloud, so that only specified cloud user (with valid decryption key) can gain access to the data via valid decryption.

To prevent shared photos being accessed by the "insiders" of the system, a straightforward way is to designate the group of authorized data users prior to encrypting the data. In some cases,

nonetheless, Alice may have no idea about who the photo receivers/users are going to be. It is possible that Alice only has knowledge of attributes w.r.t. photo receivers. In this case, traditional public key encryption (e.g., Paillier Encryption), which requires the encryptor to know who the data receiver is in advance, cannot be leveraged. Providing policy-based encryption mechanism over the outsourced photos is therefore desirable, so that Alice makes use of the mechanism to define access policy over the encrypted photos to guarantee only a group of authorized users is able to access the photos.

In a cloud-based storage service, there exists a common attack that is well-known as resource-exhaustion attack. Since a (public) cloud may not have any control over download request (namely, a service user may send unlimited numbers of download request to cloud server), a malicious service user may launch the denial-of-service (DoS)/distributed denial-of-service (DDoS) attacks to consume the resource of cloud storage service server so that the cloud service could not be able to respond honest users' service requests. As a result, in the "pay-as-you-go" model, economic aspects could be disrupted due to higher resource usage. The costs of cloud service users will rise dramatically as the attacks 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.

A. PROBLEM STATEMENT

Increasing concerns over data security and privacy in cloud environments have highlighted the need for robust access control mechanisms. Current solutions often rely on single-factor authentication or limited access controls, leading to vulnerabilities and unauthorized access. To address these challenges, there is a need for a dual access control system that combines strong authentication methods with encryption to ensure that only authorized users can access sensitive data stored in the cloud. This system should be easy to implement, scalable, and capable of protecting data from both external threats and insider attacks.

B. SCOPE OF THE PROJECT

The project focuses on improving the security of cloud-based data storage and sharing through the implementation of a dual access control system. This system will integrate robust authentication mechanisms with encryption protocols to ensure that only authorized users can access sensitive data. The project's key components include designing a scalable and secure system architecture compatible with existing cloud infrastructure. It also involves implementing multi-factor

authentication (MFA) for user verification and role-based access control (RBAC) to manage user permissions effectively. Furthermore, encryption algorithms will be applied to protect data both in transit and at rest, mitigating the risk of unauthorized access or interception. The system will be integrated with cloud services to ensure seamless and secure data access. A user-friendly interface will be developed for users to manage access permissions and perform necessary actions related to access control. Additionally, compliance with relevant regulations and standards will be ensured, with audit logs provided for monitoring access and usage. Training and documentation will also be provided to users and administrators for effective and secure use of the system. Through these measures, the project aims to enhance data security and privacy in cloud-based environments.

C. *OBJECTIVE OF THE PROJECT*

The objective of the project is to enhance the security of cloud-based data storage and sharing through the implementation of a dual access control system. This system will integrate strong authentication mechanisms with encryption protocols to ensure that only authorized users can access sensitive data. The project aims to achieve several key objectives, including improving data security by implementing robust authentication and encryption, enhancing access control through role-based access control (RBAC), and ensuring compliance with relevant regulations and standards. Additionally, the project aims to provide a user-friendly interface for managing access permissions, scalability to accommodate a large number of users and data volumes, seamless integration with existing cloud services, and comprehensive training and documentation for users and administrators. Overall, the project seeks to enhance the security and privacy of cloud-based data storage and sharing, protecting sensitive information from unauthorized access and breaches.

II. MOTIVATION

A. *Background and 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 as key policy ABE. This paper mainly deals with the former. In a CP-ABE, decryption key is associated with attribute set and ciphertext is embedded with access policy. This feature makes CP-ABE quite suitable for secure cloud data sharing
1. Hereafter, we use "dual access control" to denote the control over encrypted data and download request.
(compared to KP-ABE). Note this is so because KP-ABE requires decryption key to be associated with access policy which yields heavy storage cost for cloud user. 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 attack [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 the EDoS attack in 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 a 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 cloud is not fully considered in [38]. In this paper, we will present a new solution that requires less computation and communication cost to stand still 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 in [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, the purpose and the technique of ours are different from that of the protocols in [3], [20].

III. LITERATURE REVIEW

The literature review for the project on dual access control for cloud-based data storage and sharing reveals several key insights. Access control mechanisms in cloud computing, including Role-Based Access Control (RBAC) and Attribute-Based Access Control (ABAC), are widely used but may not provide sufficient security on their own. Authentication methods like Single Sign-On (SSO) and Multi-Factor Authentication (MFA) enhance security but may not be enough to protect against advanced threats. Encryption techniques such as AES, RSA, and ECC are essential for securing data, both at rest and in transit. However, implementing dual access control, which combines strong authentication with encryption, can significantly enhance security by adding an extra layer of protection. Challenges associated with dual access control include performance overhead and complexity, but these can be mitigated through careful design and implementation. Case studies and best practices highlight the effectiveness of dual access control in enhancing data security in cloud environments. Overall, the literature review underscores the importance of implementing dual access control to enhance the security of cloud-based data storage and sharing, especially in the face of evolving cyber threats.

IV. IMPLEMENTATION OF DUAL ACCESS CONTROL FOR CLOUD-BASED DATA STORAGE AND SHARING

The implementation of dual access control for cloud-based data storage and sharing involves several key steps to ensure the security and integrity of sensitive information. The system design phase includes architecting the framework to incorporate robust user authentication, role-based access control (RBAC), and encryption mechanisms. Multi-factor authentication (MFA) is essential to verify user identities, requiring users to provide multiple forms of verification, such as passwords, security tokens, or biometric data. RBAC is then used to assign specific roles and permissions to users, ensuring that they have access only to the data and resources necessary for their roles.

Encryption plays a crucial role in protecting data both in transit and at rest, with algorithms like AES (Advanced Encryption Standard) ensuring data security. Integration with cloud services is vital for seamless and secure access to data stored in the cloud. A user-friendly interface is designed to enable users to easily manage their access permissions, request access, and reset passwords. Compliance with relevant regulations and standards is ensured, with audit logs provided for monitoring access and usage.

Thorough testing and validation are conducted to ensure that the access control system functions as intended and is secure against various types of attacks and vulnerabilities. Continuous improvement is emphasized to address new security threats and vulnerabilities, ensuring that the dual access control system remains effective in protecting sensitive information.

A. System Architecture and Working

- **Step 1:** User Authentication: Users are required to authenticate themselves using strong authentication mechanisms, such as multi-factor authentication (MFA). This step verifies the identity of the user before they are granted access to the system.
- **Step 2:** Role-Based Access Control (RBAC): Once authenticated, users are assigned specific roles and permissions based on their role within the organization. RBAC ensures

that users have access only to the data and resources necessary for their roles.

- **Step 3:** Encryption: Data is encrypted both in transit and at rest using strong encryption algorithms, such as AES (Advanced Encryption Standard). This ensures that even if data is intercepted or accessed without authorization, it remains secure and unreadable.
- **Step 4:**Cloud Integration: The system is integrated with cloud storage and sharing services to ensure seamless and secure access to data stored in the cloud. This integration allows users to access and share data securely from anywhere, using any device.
- **Step 5:**User Interface: A user-friendly interface is provided for users to manage their access permissions, request access to additional resources, and perform other necessary actions related to access control.
- **Step 6:**Compliance and Auditability: The system ensures compliance with relevant regulations and standards regarding data security and privacy. Audit logs are maintained to monitor access and usage, providing a record of who accessed what data and when.
- **Step 7:** Scalability and Performance: The system architecture is designed to be scalable to accommodate a large number of users and data volumes without compromising performance. This ensures that the system can grow with the organization's needs.
- **Step 8:** Continuous Improvement: The system undergoes regular updates and improvements to address new security threats and vulnerabilities, ensuring that it remains effective in protecting sensitive information.

B. TECHNOLOGIES USED

1. **Creating Authentication**: Technologies such as OAuth, OpenID Connect, and SAML (Security Assertion Markup Language) can be used for user authentication. Multi-factor authentication (MFA) solutions like Google Authenticator or RSA SecurID may also be implemented.

2. **Role-Based Access Control (RBAC)**: RBAC can be implemented using frameworks and libraries such as Spring Security for Java applications or ASP.NET Identity for .NET applications. Custom RBAC implementations can also

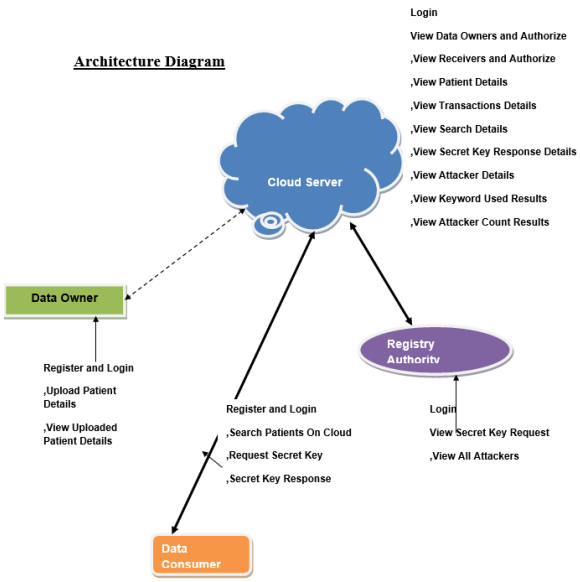

Fig. 1. System Architecture

be developed using programming languages like Python or JavaScript.

3. **Encryption**: Strong encryption algorithms such as AES (Advanced Encryption Standard) can be used to encrypt data both in transit and at rest. Libraries like OpenSSL or Bouncy Castle can be used to implement encryption in various programming languages.

4. **Cloud Services**: Integration with cloud storage and sharing services like Amazon S3, Google Cloud Storage, or Microsoft Azure Storage can be achieved using their respective SDKs and APIs.

5. **User Interface**: Web-based user interfaces can be developed using front-end technologies like HTML, CSS, and JavaScript, along with frameworks like React, Angular, or Vue.js. For mobile applications, frameworks like React Native or Flutter can be used.

6. **Compliance and Auditability**: Logging and monitoring tools such as Elasticsearch, Logstash, and Kibana (ELK stack) can be used to maintain audit logs and monitor access and usage.

7. **Scalability and Performance**: Technologies like Docker and Kubernetes can be used for containerization and orchestration to ensure scalability and performance of the system.

8. **Continuous Improvement**: Continuous inte-

gration and deployment (CI/CD) tools like Jenkins or GitLab CI/CD can be used to automate the deployment process and ensure continuous improvement of the system.

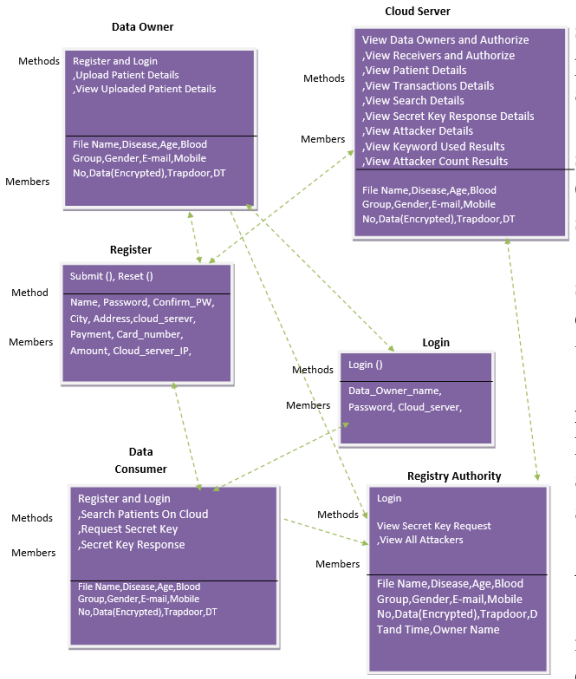

Fig. 2. A working Chart of Buddy For Bloom

C. *TOOLS AND TECHNIQUES*

Identity and Access Management (IAM) Platforms: IAM platforms such as Okta, Azure Active Directory, or AWS IAM can be used to manage user identities and access control policies.

Encryption Tools: Tools like OpenSSL, GnuPG, or Microsoft BitLocker can be used to encrypt data at rest and in transit.

Multi-Factor Authentication (MFA): MFA solutions such as Google Authenticator, Authy, or Duo Security can be used to add an extra layer of security to user authentication.

Access Control Lists (ACLs): ACLs can be used to define access control policies for specific resources or data sets.

Role-Based Access Control (RBAC): RBAC frameworks and libraries such as Spring Security (Java), ASP.NET Identity (.NET), or Django (Python) can be used to implement RBAC policies.

Logging and Monitoring Tools: Logging and monitoring tools like Elasticsearch, Logstash, and Kibana (ELK stack) can be used to monitor access and usage logs for security auditing purposes.

Data Loss Prevention (DLP) Tools: DLP tools such as Symantec DLP, McAfee DLP, or Forcepoint DLP can be used to prevent unauthorized access and sharing of sensitive data.

Secure File Sharing Solutions: Secure file sharing solutions like Box, Dropbox Business, or Google Drive Enterprise can be used to securely share files while maintaining access control.

Vulnerability Scanning Tools: Vulnerability scanning tools like Nessus, OpenVAS, or Qualys can be used to identify and mitigate security vulnerabilities in the system.

Security Information and Event Management (SIEM) Tools: SIEM tools such as Splunk, LogRhythm, or QRadar can be used to aggregate and analyze security event logs for threat detection and response.

D. *RESULT*

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 propose 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

V. 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 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 exracted. However, recent work shows that enclave may leaksome amounts of its secret(s) to a malicious host through the memory access patterns [37] or other related side-channel attacks

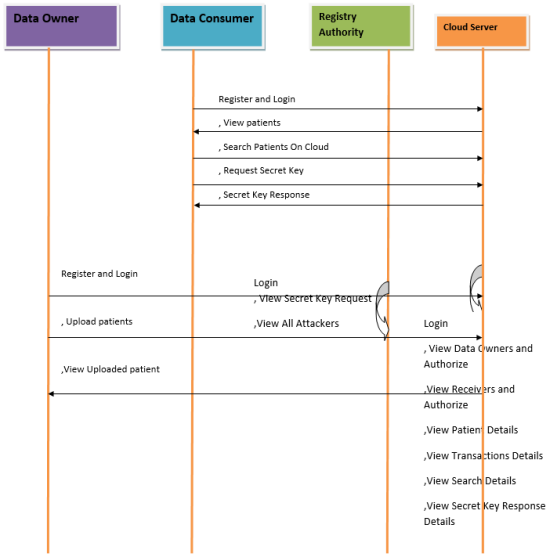

Fig. 3. Sequeance of the Buddy For Bloom

[14], [30]. The model of transparent enclave execution is hence introduced in [35]. 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

A. *Continuous Improvement of Monitoring Devices*

Continuous improvement of monitoring devices is crucial for maintaining an effective and efficient security posture. This process involves several key strategies. First, regular updates and patching ensure that devices have the latest firmware and software to address vulnerabilities and improve performance. Second, configuration management is essential to ensure devices are properly configured. Third, monitoring device performance is critical to identify and address bottlenecks or issues. Fourth, log analysis helps to detect trends, anomalies, and potential threats. Integration with SIEM systems centralizes and correlates security event data for better threat detection. Regular audits and assessments identify and address security vulnerabilities. Training personnel ensures they are aware of best practices and new threats. Incident response planning includes procedures for

responding to security incidents involving monitoring devices. Collaboration and information sharing with other organizations improve overall security posture. Finally, establishing a feedback mechanism gathers input for continuous improvement efforts. By implementing these strategies, organizations can enhance the effectiveness and efficiency of their monitoring devices, improving their ability to detect and respond to security threats.

B. Enhanced Data Analytics

Enhanced data analytics can significantly benefit organizations by providing deeper insights, improving decision-making, and driving innovation. To achieve this, organizations can implement several strategies. First, they can utilize advanced analytics techniques such as machine learning, artificial intelligence, and predictive analytics to uncover patterns and trends in data that traditional methods may miss. Second, implementing big data platforms such as Hadoop, Spark, or Google BigQuery can help store, process, and analyze large volumes of data quickly and efficiently. Third, using data visualization tools like Tableau, Power BI, or D3.js can create interactive and insightful visualizations that make complex data easy to understand.

C. Research and Publications

Research and publications are essential for advancing knowledge and driving innovation in the field of dual access control for cloud-based data storage and sharing. One key area of research is algorithm development, focusing on creating encryption algorithms and authentication mechanisms tailored for cloud environments to enhance security and minimize performance overhead. Security analysis is another crucial area, where research identifies vulnerabilities in existing dual access control systems and proposes solutions to mitigate them, improving overall security.

D. Feedback Integration

Feedback integration is essential for enhancing the effectiveness and usability of dual access control systems for cloud-based data storage and sharing. One key aspect of feedback integration is implementing a user feedback mechanism within the access control system. This allows users to provide feedback on their experience, including ease of use, performance, and any issues encountered. Additionally, conducting surveys, interviews, and usability testing with users can provide valuable insights into areas for improvement.

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 SP 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 Serge 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