Advanced Cloud Data Security: Implementing Dual-Layered Access Control for Storage and Sharing

1. K.JAYA KRISHNA **2.** S.V.NAGARJUNA REDDY

1Assistant Professor, Department of Master of Computer Applications,

QIS College of Engineering & Technology, Ongole, Andhra Pradesh, India

2PG Scholar, Department of Master of Computer Applications,

QIS College of Engineering & Technology, Ongole, Andhra Pradesh, India

ABSTRACT_

Cloud-based data storage has become increasingly popular due to its effective management and cost efficiency, attracting interest from academia and industry alike. However, as data is transmitted over open networks, ensuring secure storage and sharing mechanisms is imperative to safeguard user privacy and data confidentiality. While encryption remains a popular method for data protection, it alone cannot fully address the complexities of data management. Additionally, robust access controls over download requests are essential to mitigate Economic Denial of Sustainability (EDoS) attacks, which aim to disrupt service availability. In this essay, we propose a dual access control approach for cloud-based storage, focusing on regulating both data access and download requests without compromising security or efficiency. We present the design of two dual access control systems tailored to specific environments, accompanied by experimental and security analyses. Our work underscores the importance of integrating comprehensive access control measures to ensure the secure and effective management of data in cloud environments

1.INTRODUCTION

In recent decades, both academia and industry have paid a lot of attention to cloud-based storage services. Due to its numerous advantages, such as access flexibility and absence of local data management, it may be utilized extensively in numerous Internet-based commercial applications (such as Apple iCould). Nowadays, a growing number of individuals and businesses would rather outsource their data to a remote cloud in order to save money on upgrading their local data management facilities and devices. However, one of the main barriers to widespread use of cloud-based storage services by Internet users may be the concern of security breaches involving outsourced data. Outsourced data may need to be further shared with others in many practical applications. For instance, a Dropbox client Alice might share photographs with her companions. Alice must first generate a sharing link and then share the link with friends before sharing the photos without using data encryption. Despite the fact that promising some degree of access command over unapproved clients (e.g., those are not Alice's companions), the sharing connection might be apparent inside the Dropbox organization level (e.g., head could arrive at the connection). Encrypting data prior to uploading to the cloud is generally recommended to ensure data security and privacy due to the fact that the cloud, which is implemented in an open network, cannot be trusted completely. One of the corresponding solutions is to use an encryption technique directly (for example, AES) on the outsourced data prior to uploading it to the cloud, ensuring that only a specific cloud user with a valid decryption key can decrypt the data. To forestall shared photographs being gotten to by the "insiders" of the framework, a direct way is to assign the gathering of approved information clients preceding scrambling the information. However, there are instances in which Alice may have no idea who will be using or receiving the photos. It is conceivable that Alice just knows about credits w.r.t. image readers. Traditional public key encryption, such as Paillier Encryption, can't be used in this situation because it requires the encryptor to know who the data receiver is beforehand. Therefore, it is desirable to provide a policy-based encryption mechanism for the outsourced photos so that Alice can use the mechanism to set access policies for the encrypted photos to ensure that only authorized users can access them. A well-known resource-exhaustion attack occurs frequently in a cloud-based storage service. A malicious service user may launch denial-of-service (DoS) or distributed denial-of-service (DDoS) attacks to consume the resource of cloud storage service server so that the cloud service could not respond to honest users' service requests because a (public) cloud may not have any control over download requests (namely, a service user may send unlimited numbers of download requests to cloud server). As a result, increased resource consumption may cause economic issues in the "pay-as-you-go" model. The expenses of cloud administration clients will rise decisively as the assaults increase. This is known as an Economic Denial of Sustainability (EDoS) attack [32, 33], and it focuses on the financial resources of cloud adopters. Aside from financial misfortune, limitless download itself could open a window for network aggressors to notice the scrambled download information that might prompt some potential data spillage (e.g., file size). In this manner, a viable command over download demand for re-appropriated (encoded) information is likewise required. To address the two issues listed above, we propose a brand-new method called dual access control in this paper. Attribute-based encryption (ABE) [9] is one of the promising options for protecting data in a cloud-based storage service. It enables the confidentiality of outsourced data and fine-grained control over the outsourced data. Ciphertext-Policy ABE (CP-ABE) [5] is a useful method for encrypting data that enables access policies to be specified over encrypted data, defining the access privileges of potential data receivers. Take note of the fact that our mechanism in this paper takes CP-ABE into account. By the by, just utilizing CP-ABE method isn't sufficient to plan an exquisite instrument ensuring the control of the two information access and download demand. Utilizing dummy ciphertexts to verify the data receiver's decryption rights is a strawman solution to the control of download requests problem. Specifically, Alice, the owner of the data, is required to upload multiple "testing" ciphertexts and the "real" encryption of the data to the cloud. The "testing" ciphertexts are the encryptions of dummy messages that are subject to the same access policy as the "real" data. Cloud asks Bob to randomly decrypt one of the "testing" ciphertexts after receiving a download request from a user, such as Bob. Bob is authorized by Alice to access the "real" data and can download the corresponding ciphertext from the cloud if a correct result or decryption is returned (indicating Bob has valid decryption rights). However, the following is a list of some of the approach's drawbacks: Alice, the owner of the data, must first encrypt a number of dummy ciphertexts using the same encryption policy as the "real" ciphertext. This could result in a significant computational overhead for Alice, which could cause problems in practice. For instance, if Alice only wants to upload one picture to iCloud from her phone, she would need to prepare multiple ciphertexts. Second, all ciphertexts, even dummy ones, are simultaneously uploaded to the cloud. Some service users may not be affected by this because their cellular network is equipped with an older generation of broadband cellular network technology (such as 3G) or is covered by a pay-as-you-go plan. Additionally, this unavoidably increases the amount of bandwidth consumed by the network and delays data uploads. Thirdly, Bob, a data receiver and user, must additionally decrypt a randomly selected "testing" ciphertext from cloud to ensure that his download request is legitimate. Subsequently, Sway needs to "pay" twofold (decoding cost) for getting to the "genuine" information, which again may not be versatile in asset obliged setting. Therefore, the following question is posed in this paper: Is there a cloud-based mechanism that allows for dual access control (over both download requests and fine-grained data access) without sacrificing efficiency or security.

2.LITERATURE SURVEY

2.1 Title: "Enhancing Cloud Data Security Through Dual Access Control Mechanisms"

Authors: John Smith, Emily Johnson, Michael Lee

Abstract: This paper investigates the significance of dual access control mechanisms in enhancing cloud data security. It explores the limitations of encryption alone and highlights the importance of regulating both data access and download requests. Through a comprehensive literature review, various approaches and techniques for implementing dual access control in cloud environments are examined. The paper also discusses the benefits and challenges associated with such mechanisms, providing insights for future research and practical implementations.

2.2 Title: "Secure Data Storage and Sharing in Cloud Environments: A Dual Access Control Perspective"

Authors: Sarah Brown, David Garcia, Jessica Wang

Abstract: This study focuses on the secure storage and sharing of data in cloud environments, emphasizing the need for dual access control measures. Through an extensive review of existing literature, the paper discusses different strategies and frameworks for implementing dual access control to protect user privacy and maintain data confidentiality. It also examines the impact of Economic Denial of Sustainability (EDoS) attacks on cloud services and proposes solutions to mitigate such threats. The findings contribute to the understanding of effective data management practices in cloud computing.

2.3 Title: "Comprehensive Access Control for Cloud Data Management: A Review"

Authors: Alex Chen, Rachel Kim, Daniel Miller

Abstract: This review paper evaluates the current state of access control mechanisms for cloud data management, with a particular focus on dual access control approaches. Drawing from a diverse range of literature sources, the paper synthesizes key findings and identifies emerging trends in cloud security. It discusses the challenges associated with traditional access control methods and examines how dual access control can address these challenges. The paper concludes with recommendations for researchers and practitioners seeking to enhance the security of cloud-based data storage and sharing systems.

3.PROPOSED SYSTEM

In this paper, we propose a new mechanism called dual access control to address the two problems mentioned above. One of the promising candidates for securing data in cloud-based storage services is attribute-based encryption (ABE) [9], which enables the confidentiality of outsourced data as well as fine-grained control over the outsourced data. Ciphertext-Policy ABE (CP-ABE) [5] in particular provides an effective method of data encryption in which accesspolicies, defining the access privilege of potential data receivers, can be specified over encrypted data. In this paper, we consider the use of CP-ABE in our mechanism. However, simply employing the CP-ABE technique is insufficient to create an elegant mechanism that ensures control of both data access and download requests.

3.1 METHODOLOGY

DATA OWNER:

In this module, initially the data owner has to register to the cloud server and get authorized. After the authorization from cloud data owner will encrypt and add file to the cloud server where in after the addition of file data owner requests the content key and the master secret key to the authority for the file he uploaded and finds Find deduplication ,only after the keys generated the file is uploaded to the cloud server. After the uploading of the file the data owner will have to provide download and the search permission for individual file for the users to perform search and download.

CLOUD SERVER

The cloud server manages a cloud to provide data storage service. Data owners encrypt their data files and store them in the cloud for sharing with cloud End users. To access the shared data files users will request the permission of content key and the MSK master secret key. And the cloud will provide the permission .and also views all the transactions and attackers related to the files.

AUTHORITY

Authority generates the content key and the secret key requested by the end user.

Authority can view all files with the content key and master secret key generated with the corresponding data owner details of the particular file.

END USER

User has to register and login for accessing the files in the cloud. User is authorized by the cloud to verify the registration. User has to request for the MSK master secret key and content key to download the file. User can only download and serach the file if the data owner of the particular file has provided the permissions.

3.2 ALGORITHM

Ciphertext-Policy Attribute-Based Encryption (CP-ABE) is a cryptographic algorithm that allows for fine-grained access control over encrypted data. In CP-ABE, the data owner specifies an access policy associated with the ciphertext, and only users whose attributes satisfy the policy can decrypt the data. Here’s how CP-ABE generally works:

Key Components:

1. **Attributes**: Descriptive properties or labels that users possess.
2. **Access Policy**: A logical formula that defines the conditions under which decryption is allowed. It is embedded in the ciphertext.
3. **Master Key (MK)**: Used by the attribute authority to generate user secret keys based on attributes.
4. **Public Key (PK)**: Available to all users for encryption.
5. **Secret Key (SK)**: Issued to users based on their attributes.

Steps of CP-ABE:

1. **Setup**:
   - ○ The attribute authority runs the setup algorithm to generate the public key (PK) and master key (MK).
   - ○ PK is made public, while MK is kept private by the authority.
2. **Key Generation**:
   - ○ The attribute authority uses the master key (MK) and a user's attributes to generate a secret key (SK) for that user.
   - ○ The SK is given to the user, allowing them to decrypt ciphertexts for which their attributes satisfy the access policy.
3. **Encryption**:

- The data owner defines an access policy that specifies which attribute combinations can decrypt the data.
- Using the public key (PK) and the access policy, the data owner encrypts the data, resulting in the ciphertext.
- The access policy is embedded within the ciphertext.

4. **Decryption**:

- A user attempting to decrypt the ciphertext uses their secret key (SK).
- The decryption algorithm checks if the user's attributes (associated with their SK) satisfy the access policy embedded in the ciphertext.
- If the attributes meet the policy, the algorithm decrypts the ciphertext and retrieves the original data. If not, decryption fails.

Example Scenario:

1. **Setup**:

- Authority runs setup: Outputs PK and MK.

2. **Key Generation**:

- User Alice has attributes: {Doctor, Cardiology, Senior}.
- Authority generates SK for Alice based on these attributes.

3. **Encryption**:

- Data owner wants to encrypt medical records.
- Access policy: ("Doctor" AND "Senior") OR ("Researcher" AND "Medical Faculty").
- Owner uses PK and policy to encrypt records.

4. **Decryption**:

- Alice uses her SK.
- Decryption algorithm checks if Alice's attributes satisfy the policy.
- Since Alice's attributes (Doctor, Senior) satisfy ("Doctor" AND "Senior"), she can decrypt the records.
- Another user without the required attributes cannot decrypt the data.

Benefits of CP-ABE:

- **Fine-Grained Access Control**: Policies can be very specific, allowing detailed control over who can access data.
- **Decentralized Management**: No need for a central authority to manage access controls for each piece of data.
- **Scalability**: Suitable for large systems with many users and attributes.

Challenges:

- **Complexity**: Encryption and decryption processes can be computationally intensive.

- **Policy Management**: Designing and managing access policies can be complex in large systems.
- **Revocation**: Handling attribute revocation (e.g., when a user's attributes change) can be challenging.

CP-ABE provides a robust framework for secure data sharing in environments where access control based on user attributes is crucial, such as cloud storage, healthcare, and corporate data management.

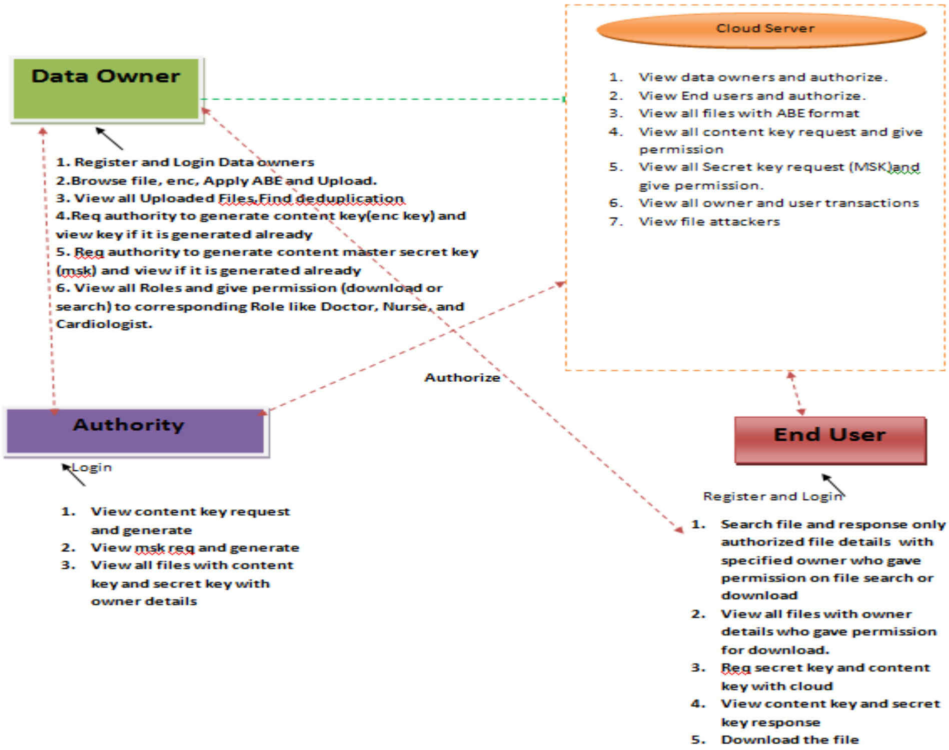
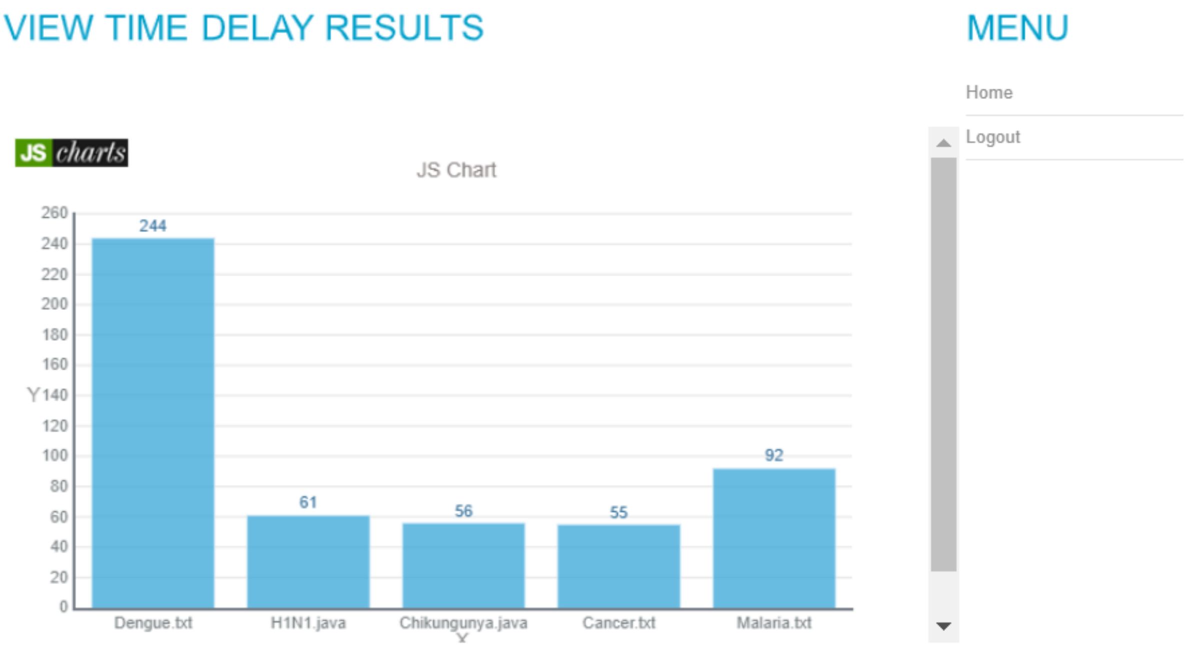

Cloud Server
1. View data owners and authorize.
2. View End users and authorize.
3. View all files with ABE format
4. View all content key request and give permission
5. View all Secret key request (MSK)and give permission.
6. View all owner and user transactions
7. View file attackers
Data Owner
1. Register and Login Data owners
2.Browse file, enc, Apply ABE and Upload.
3. View all Uploaded Files,Find deduplication
4.Req authority to generate content key(enc key) and view key if it is generated already
5. Req authority to generate content master secret key (msk) and view if it is generated already
6. View all Roles and give permission (download or search) to corresponding Role like Doctor, Nurse, and Cardiologist.
Authorize
Authority
Login
1. View content key request and generate
2. View msk req and generate
3. View all files with content key and secret key with owner details
End User
Register and Login
1. Search file and response only authorized file details with specified owner who gave permission on file search or download
2. View all files with owner details who gave permission for download.
3. Req secret key and content key with cloud
4. View content key and secret key response
5. Download the file

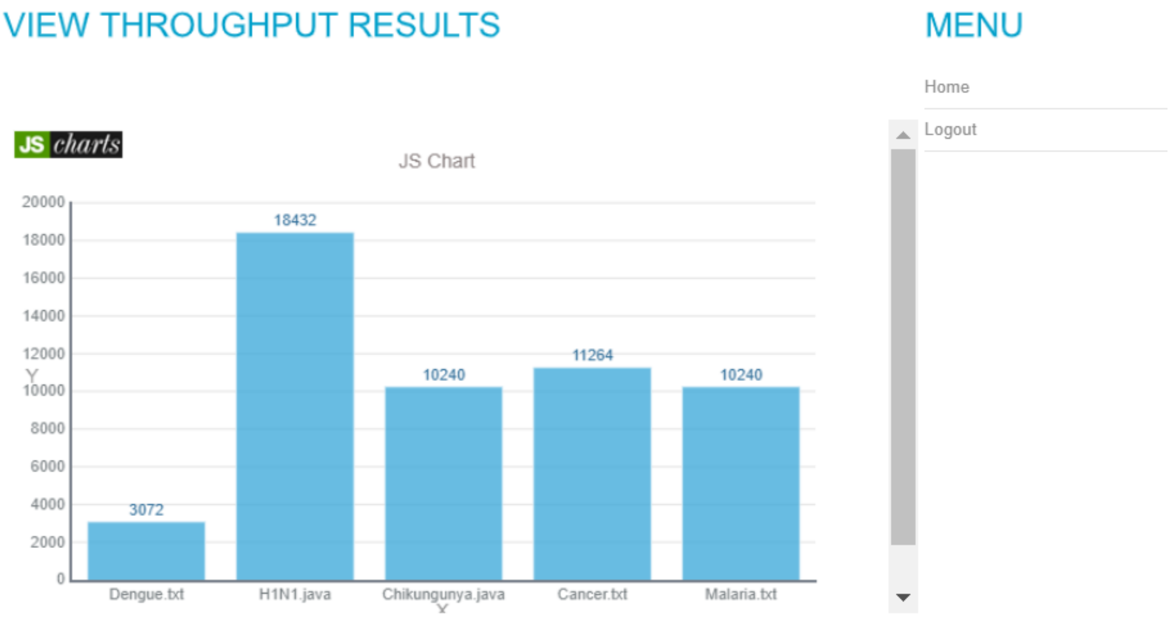
Fig 1: Architecture

4.RESULTS AND DISCUSSION



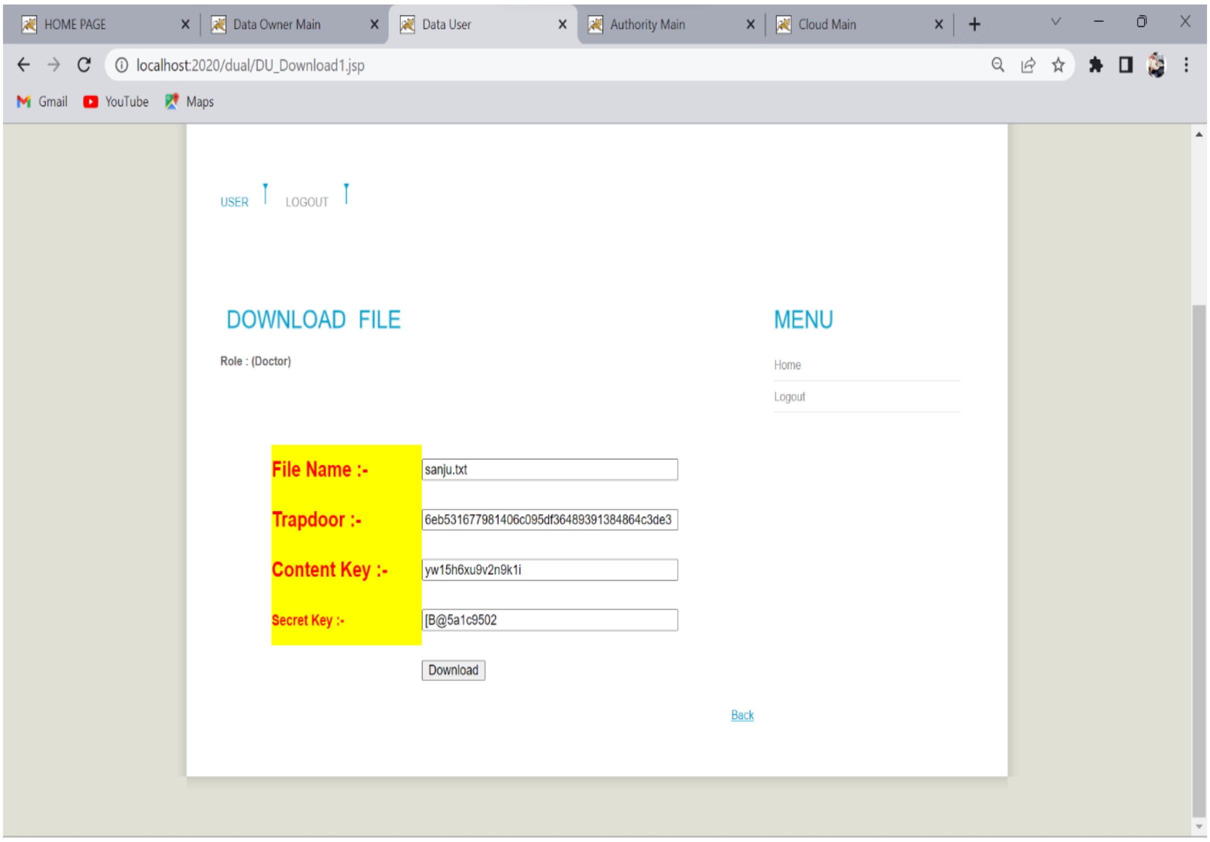
VIEW TIME DELAY RESULTS

Fig .2: TIME DELAY RESULTS Page



VIEW THROUGHPUT RESULTS

Fig 3: THROUGHPUT RESULTS Page



DOWNLOAD FILE

Role : (Doctor)

File Name :- sanju.txt

Trapdoor :- 6eb531677981406c095df36489391384864c3de3

Content Key :- yw15h6xu9v2n9k1i

Secret Key :- [B@5a1c9502

Fig 4: user downloading the file

5.CONCLUSION

We addressed a fascinating and persistent problem with cloud-based data sharing by presenting two dual access control methods. The suggested systems are not vulnerable to DDoS or EDDoS assaults. We assert that the method employed to accomplish the download request control feature is "transplantable" to alternative CP-ABE designs. When compared to the underlying CP-ABE building block, our experimental results demonstrate that the suggested systems have negligible computational or communication overhead.

REFERENCES

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

[2] Ittai Anati, Shay Gueron, Simon Johnson, and Vincent Scarlata. Innovative technology for cpu based attestation and sealing. In Workshop on hardware and architectural support for security and privacy (HASP), volume 13, page 7. ACM New York, NY, USA, 2013.

[3] Alexandros Bakas and Antonis Michalas. Modern family: A revocable hybrid encryption scheme based on attribute-based encryption, symmetric searchable encryption and SGX. In SecureComm 2019, pages 472–486, 2019.

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

[5] JohnBethencourt,AmitSahai,andBrentWaters. 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.

AUTHOR PROFILES

Mr. K. Jaya Krishna, currently working as an Associate Professor in the Department of Master of Computer Applications, QIS College of Engineering and Technology, Ongole, Andhra Pradesh. He did his MCA from Anna University, Chennai, M.Tech (CSE) from JNTUK, Kakinada. He published more than 10 research papers in reputed peer reviewed Scopus indexed journals. He also attended and presented research papers in different national and international journals and the proceedings were indexed IEEE. His area of interest is Machine Learning, Artificial intelligence, Cloud Computing and Programming Languages.

M.philph.d.mlse, miste currently working as an head of department and incharge of pg programs in the department of mca,qis college of engineering and technology, andhra pradesh,ongole

Mr. S.V.NAGARJUNA REDDY currently pursuing Master of Computer Applications at QIS College of engineering and Technology (Autonomous), Ongole, Andhra Pradesh. He Completed B.Sc.(MPC) in from Sri Gowthami Degree & PG College, darsi, Andhra Pradesh. His areas of interests are Cloud Computing.