IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

Secure Cloud Storage Against Data with Sanitizable Access Control System

[1]Izhar Idrisi, [2]Mustaque Ahamad, [3]Aamir Momin, [4]Prof. Rinkal Bari
[1 2 3]UG student, [4]Assistant Professor
[1 2 3 4]Department of Information Technology,
[1 2 3 4]Theem College of Engineering, Boisar, India

Abstract: Cloud computing stands as a pivotal component within the IT industry, promising substantial cost reductions in hardware and software resources. It facilitates seamless data sharing among corporate employees, predominantly leveraging cloud storage. Despite the convenience of storing data as plain text with access controls, relying solely on the cloud's trustworthiness, given its third-party ownership, proves impractical. Encryption thus becomes imperative, mandating data to be stored as cipher text with stringent access controls. However, the presence of malicious insiders who may adhere to sharing policies yet create vulnerable cipher texts poses a formidable challenge. Existing literature predominantly concentrates on ensuring legitimate recipients can decrypt data stored in the cloud, neglecting issues stemming from malicious data publishers. These individuals comply with sharing policies but craft cipher texts susceptible to unauthorized decryption, posing a significant threat to corporate intellectual property. To bridge this gap, we introduce the concept of a Sanitizable Access Control System (SACS) aimed at fortifying cloud storage against malicious data publishers. SACS represents a pioneering approach to access control management, effectively mitigating risks associated with malevolent actors. This research direction offers a practical solution for safeguarding data integrity and confidentiality in cloud environments, underscoring the criticality of addressing emergent threats within cloud computing security. In essence, SACS serves as a robust mechanism to thwart potential breaches stemming from malicious insiders, thereby bolstering the security posture of cloud storage systems and ensuring the protection of sensitive corporate data**.**

Keywords—Cloud Computing, Encryption, Access Control, Receivers, Ciphers, Servers, Security, Secure Cloud Storage, Access Control, Sanitizable, Malicious Data Publisher.

I. INTRODUCTION

Cloud storage has profoundly transformed enterprise operations, particularly benefiting Small and Medium-sized Enterprises (SMEs) with its cost-effective solutions. However, relying solely on plaintext storage and access controls in the cloud is impractical due to the inherent risk of potential data leaks. While Attribute-based Encryption (ABE) has been utilized to safeguard data, it proves inadequate in addressing the threat posed by malicious data publishers who may encrypt data in a manner that facilitates unauthorized access. To tackle this challenge, the proposed Sanitizable Access Control System (SACS) offers a pragmatic solution. SACS introduces a flexible access control mechanism for both data publishers and receivers, akin to ABE, but with an added sanitizing capability. This feature prevents malicious data publishers from generating decryptable ciphertexts without possessing valid private keys, thereby ensuring data privacy even in the presence of malicious actors. The motivation behind this endeavor is to address the pressing challenge posed by malicious data publishers within cloud storage environments. While cloud technology has revolutionized enterprise operations, particularly benefiting SMEs with its low-cost solutions, the assumption of complete trust in the cloud is no longer feasible.

Encryption becomes imperative to protect sensitive data from potential breaches. However, traditional methods like ABE fall short when dealing with malicious insiders who may intentionally leak sensitive information, posing a significant threat to data privacy and security. The proposed SACS aims to fill this gap by incorporating a sanitizing capability to thwart malicious data publishers. By providing flexible access control and integrating a mechanism to prevent the generation of decryptable cipher texts without valid private keys, SACS offers a practical solution to safeguard data integrity and confidentiality in cloud storage environments. The scope of this project is to address the challenge of ensuring data privacy and security in cloud storage, particularly in the face of malicious data publishers. By introducing SACS, which provides flexible access control and includes a sanitizing capability, the proposed solution mitigates the risks associated with malicious actors, thereby enhancing the overall security posture of cloud storage systems.

II. LITERATURE SURVEY

A systematic literature review is a means of evaluating and interpreting all available research relevant to a particular research question, topic, or phenomenon of interest. The scientific databases with full-text papers and the other available scientific articles in the field of social sciences were used in the research. All scientific and other papers and works written in the period from 2009 to March 2020 are taken into account.

Sr. No.	Paper Title [Reference]	Author Name	Advantages	Disadvantages
[A]	Attribute-based encryption for fine-grained access control of encrypted data	V. Goyal, O. Pandey	As more sensitive data is shared and stored by third-party sites on the Internet, there will be a need to encrypt data stored at these sites	Need for specialized consultants andtools for implementing .
[B]	Ciphertext-policy attribute-based encryption	J. Bethencourt	In several distributed systems a user should only be able to access data if a user possesses a certain set of credentials or attributes.	Not focused onnegative aspect ofthe system.
[C]	Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization	B. Waters	The only previous work to achieve these parameters was limited to a proof in the generic group model. We present three constructions within our framework.	Considered only two public sectors.

[D]	Security intelligence for cloud management infrastructures	S. Berger et al.	We address the problem of protecting cloud infrastructures and customer workloads via smart auditing and logging & satisfying.	Provides a basic system which is existing.
[E]	Improved proxy re-encryption schemes with applications to secure distributed storage	G. Ateniese, K. Fu, M. Green, and S. Hohenberger	Proxy re-encryption adds access control to encrypted file systems efficiently	Adoption hindered by security risks; newer schemes offer stronger security but may require further validation.

III. SYSTEM ARCHITECTURE

In the Secure Access Control System (SACS), a trusted authority manages the master secret key and issues unique private keys to registered receivers. Data publishers encrypt plain data with a key and set access policies. Sanitizers transform cipher data, which is stored in the cloud for receivers to access. Receivers request private keys from the authority to decrypt data. The cloud server stores and provides cipher data without computation, regardless of its behavior.

A. Design

SACS enhances data privacy by sanitizing cipher data, preventing malicious behaviors that lead to invalid access. It ensures data integrity by checking if cipher data adheres to claimed access policies before sanitizing. Additionally, SACS enforces stronger access control, allowing only valid receivers to decrypt plain data; even if possessing an encryption key from a malicious data publisher, receivers cannot decrypt sanitized cipher data correctly.

B.Requirement Analysis

In the software development lifecycle, demand analysis is one of the most important phases. It's used to identify and define the software. For any software design, there are different kinds of conditions to be fulfilled to insure the smooth handling of the processes. easily defined conditions are important labels on the road to a successful design.

Table 1. Requirements of ERP System

Software Requirements	Hardware Requirements
Java Development Kit	Windows 10 Pro
NetBeans	8 GB RAM
SQL	Intel(R) Core(TM) i5 1.60GHz 1.80 GHz Processor
Cloud	Wi-Fi Router
Apache Tomcat	100GB free Hard Disk

C.Proposed System

Our primary objective is to ensure data privacy, particularly in scenarios where data publishers act maliciously and deviate from encryption protocols. To address this, we propose a practical solution known as the Sanitizable Access Control System (SACS), specifically designed for cloud storage to counteract malicious data publishers. SACS facilitates flexible access control for both data publishers and receivers. Key to SACS is its sanitizing capability, which prevents malicious data publishers from generating ciphertexts that could be decrypted without valid private keys. Even if malicious actors produce ciphertexts that are decryptable by anyone, SACS intervenes by transforming these ciphertexts into new ones that are only decryptable by valid private key holders. We outline the architecture and scheme necessary to realize this concept and provide an implementation of SACS. In our framework, the entity sending the cipher data is referred to as the data publisher, while the entity retrieving the plain data is termed the receiver, with the cloud serving as the storage platform. SACS aims to offer flexible access control for both data receivers and publishers. It restricts access to plain data solely to valid data receivers possessing private keys issued by a trusted authority. Through ciphertext sanitization, SACS effectively prevents malicious data publishers from producing information capable of retrieving decryption keys without valid private keys generated from the trusted center, such as encryption keys. Consequently, even if unauthorized receivers possess encryption keys, they cannot access plain data due to SACS's security measures.

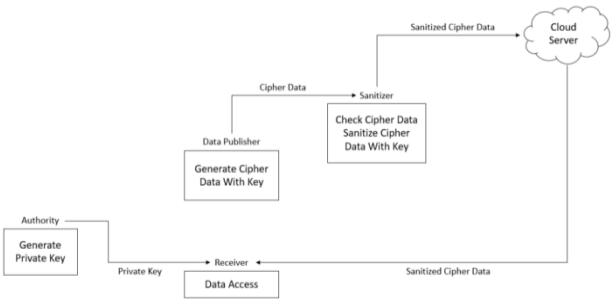

Fig 1. Proposed system of secure cloud storage against data with a sanitizable access control system

Our main thing is to achieve data sequestration when data publishers are vicious and they don't follow the encryption algorithm consequently. We aim to propose a veritably practical notion, called a Sanitizable Access Control System, or simply SACS, which is designed for pall storehouse to repel vicious data publishers. SACS enables a flexible access control for both data publishers and data receivers.

D. System Design

System design is the process of planning system elements similar as armature, modules, and factors, the colourful interfaces of these factors, and the data passing through the system. The thing of the system design process is to give sufficient detailed information and knowledge. information about the system and its system elements so that the perpetration is compatible with the architectural units defined in the models and views of the system architecture.

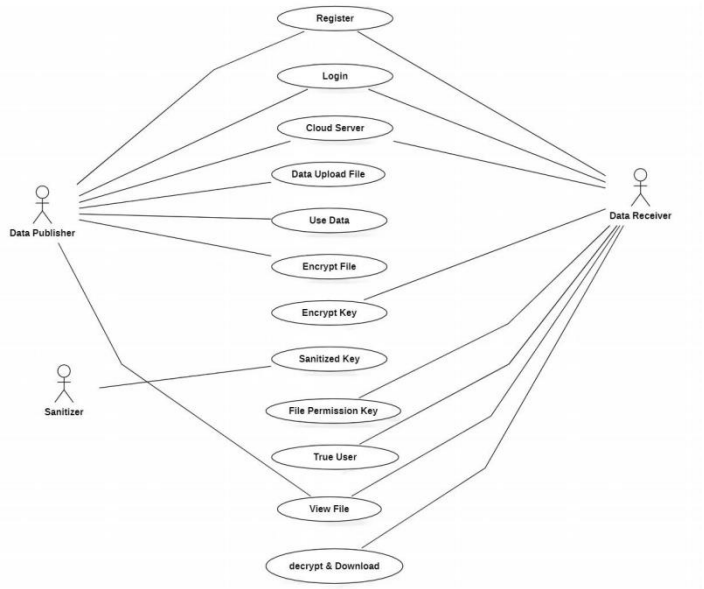

Fig 2. Use case of Secure cloud storage against data with a sanitizable access control system

E. Data Flow

A data flow diagram represents the data flow of a process or system usually an information system. A data flow diagram has no control flow - it has no decision rules and no loops. A data flow diagram (DFD) is a graphic or visual representation that uses a standardized set of symbols and notations to describe the operation of a business through the transmission of information. It gives a more clear idea of our project. It expands on each process to give detailed information about the process.

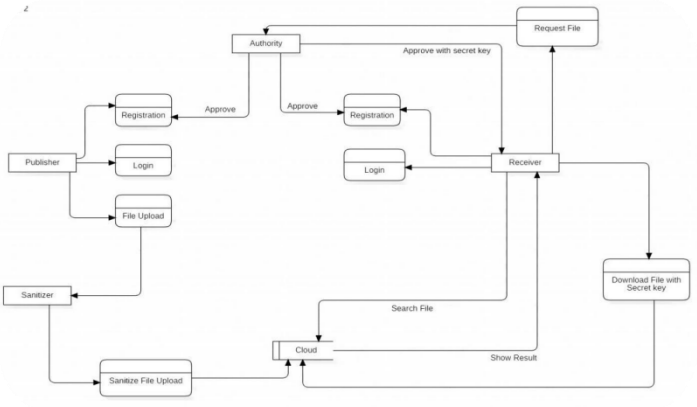

Fig 3. Data Flow of Secure cloud storage against data with a sanitizable access control system

IV. RESULTS

The Sanitizable Access Control System (SACS) for Secure Cloud Storage effectively mitigates the risks posed by malicious data publishers. By employing a trusted authority to manage encryption keys and issuing unique private keys to registered receivers, SACS establishes a secure framework for data access. Publishers encrypt data and define access policies, while sanitizers ensure that only authorized receivers can decrypt and access the data. This process prevents unauthorized access, even if publishers behave maliciously by distributing encryption keys to non-registered entities. SACS operates transparently, with the sanitizer executing a predefined sanitization algorithm to transform cipher data without compromising the integrity or confidentiality of the original data. Receivers can securely download cipher data from the cloud server, register, and obtain private keys from the authority to access plain data, ensuring that only valid receivers with the necessary permissions can decrypt and access the information.

Fig 4. Home Page

The image depicts the homepage of a system architecture involving various entities: Authority, Data Publisher, Sanitizer, Receiver, and Cloud Server. These entities likely interact within a secure cloud storage system. The Authority likely manages access control policies, the Data Publisher uploads encrypted data, the Sanitizer processes data to ensure integrity, the Receiver accesses authorized data, and the Cloud Server hosts the encrypted data.

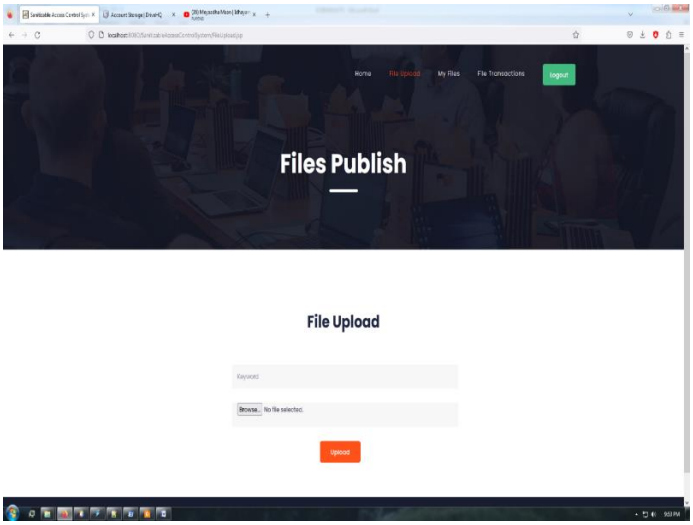

Fig 5. File Publish Page

The file publish page depicted in the image serves as the interface for data publishers to upload files to the secure cloud storage system. Publishers can select files from their local devices and initiate the upload process. The publisher also provides a keyword for accessing the file, enabling the receiver to access it using that keyword.

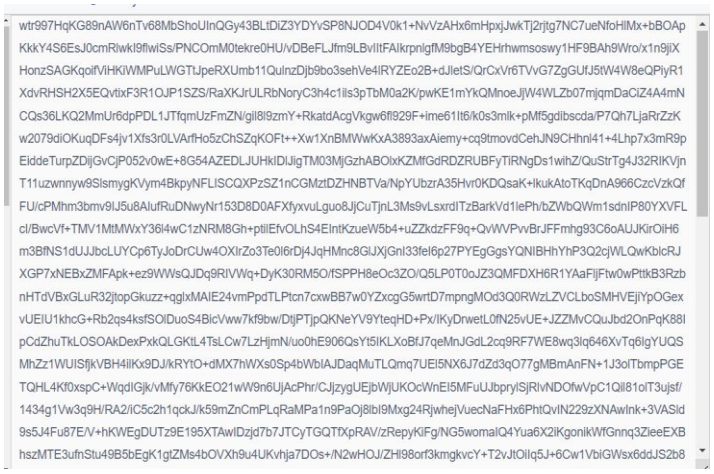

Fig 6. Encrypted File

Encrypt files using Sanitizable access control, ensuring sensitive data remains secure. This method allows users to specify who can read, write, or modify the file, enhancing confidentiality and integrity. By implementing tailored access permissions, organizations can safeguard their information from unauthorized access or tampering. This approach provides granular control over data, mitigating risks and ensuring compliance with privacy regulations.

File Id	Data Publisher Id	Data Publisher Name	File Name	Uploaded Time
1	1	zack	GET JD 2024.pdf	2024/02/16 11:00:25
2	1	zack	resume..docx	2024/02/16 11:07:49
3	1	zack	result.pdf	2024/02/16 11:10:00
4	1	zack	abstract.txt	2024/02/16 14:40:37

Fig 7 Uploaded File on Cloud

The displayed image showcases a file uploaded to the cloud storage system, providing essential details such as the file ID, publisher's name, file name, publisher ID, and the date and time of upload. This information enables effective tracking and management of files within the system, ensuring accountability and facilitating efficient retrieval when needed.

V. CONCLUSION

We initiated the study of secure cloud storage in the presence of malicious data publishers, which is a very practical situation that unfortunately has never been studied in the literature previously. In this setting, malicious data publishers construct data following the given access control policy, but the ciphertexts can be decrypted by unauthorized users without the need for valid keys. We designed a system and its secure scheme to enable protection against this kind of attack. We also provided an implementation of our system for performance analysis. We believe this work will open future research work in cloud storage since this notion is very practical. We note that this notion will further encourage the adoption of cloud storage in practice.

VI. ACKNOWLEDGMENT

We'd like to express our sincere gratefulness to all those who contributed to the successful completion of this design. First and foremost, we extend our deepest appreciation to our design administrator Prof. Rinkal Bari whose guidance, support, and inestimable perceptivity have been necessary throughout the entire duration of this design. Their moxie and stimulant have been necessary in steering us in the right direction and prostrating colorful challenges along the way. We're also immensely thankful to the entire platoon involved in the design, whose fidelity, collaboration, and hard work have been vital in bringing this vision to consummation. Each platoon member's unique chops and benefactions have played a pivotal part in the development, perpetration, and testing phases of the project.

REFERENCES

[1] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in Proc. 13th ACM Conf. Comput. Commun. Secur., 2006, pp. 89–98.
[2] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in Proc. IEEE Symp. Secure. Privacy, 2007, pp. 321–334.
[3] B. Waters, “Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization,” in Proc. Int. Workshop Public Key Cryptogr., 2011, pp. 53–70.
[4] S. Berger et al., “Security intelligence for cloud management infrastructures,” IBM J. Res. Develop., vol. 60, no. 4, pp. 11:1–11:13, 2016.
[5] D. Boneh, C. Gentry, and B. Waters, “Collusion resistant broadcast encryption with short ciphertexts and private keys,” in Proc. Annu. Int. Cryptol. Conf., 2005, pp. 258–275.