Securing The Cloud Storage Against Malicious Data Publishers

R Deepika[1], G Balasowri[1], VK Sathya Priya[1], G Chaitanya[1],
K Yogesh[1], Sivasankar Chittoor[2]

[1] *Research Scholar, Department of Computer Science & Information Technology, Siddharth Institute of Engineering & Technology, Andhra Pradesh, India*
[2] *Assistant Professor, Department of Computer Science & Information Technology, Siddharth Institute of Engineering & Technology, Andhra Pradesh, India*

ABSTRACT

Cloud computing is considered as one of the most prominent paradigms in the information technology industry, since it can significantly reduce the costs of hardware and software resources in computing infrastructure. This convenience has enabled corporations to efficiently use the cloud storage as a mechanism to share data among their employees. At the first sight, by merely storing the shared data as plaintext in the cloud storage and protect them using an appropriate access control would be a nice solution. This is assuming that the cloud is fully trusted for not leaking any information, which is impractical as the cloud is owned by a third party. Therefore, encryption is mandatory, and the shared data will need to be stored as a ciphertext using an appropriate access control. However, in practice, some of these employees may be malicious and may want to deviate from the required sharing policy. The existing protection in the literature has been explored to allow only legitimate recipients to decrypt the contents stored in the cloud storage, but unfortunately, no existing work deals with issues raised due to the presence of malicious data publishers. Therefore, it remains an elusive research problem on how to enable a sound approach to resolve the issue when malicious data publishers are involved in the system, which is a very practical question. In this work, present a new direction of research that can cope with the presence of malicious data publishers. resolve the aforementioned problem by proposing the notion of Sanitizable Access Control System (SACS), which is designed for a secure cloud storage that can also resist against malicious data publishers.

Keywords—*Data Publishers, Sanitization, Malicious, Cloud, Access control, Private key, Public key, Secret key.*

1. INTRODUCTION

The advent of cloud computing constituted one of the most momentous transformations, signaling the start of the digital era. Since its creation, technology has had a big impact on how businesses operate. Small and medium-sized enterprises (SM-Es) greatly benefit from inexpensive cloud storage choices. Thanks to cloud storage, businesses can easily share their information with their workers. One of the most important changes ushering in the digital era was the development of cloud computing. Technology has had a significant influence on business operations since its inception. Small and medium-sized businesses (SM-Es) greatly benefit from affordable cloud storage options. Businesses may readily exchange information with their employees thanks to cloud storage. The creation of cloud computing was one of the biggest shifts ushering in the digital era. Since its creation, technology has had a big impact on how businesses operate. Affordable cloud storage options are extremely beneficial to small and medium-sized enterprises (SM-Es). Thanks to cloud storage, companies can easily share information with their staff. In the body of recent research, attribute-based encryption (A-BE) is used to protect the data with appropriate access control, allowing such an illegal prevention. Any person with a valid encryption key that complies with the access rules can accurately decode the data.

2. LITERATURE SURVEY

In this section, we review some closely related work in the literature.

Access Control : Access control is able to guarantee data security in cloud storage systems. This has attracted much attention from academia and industry. IBM developed the capability-based model and systematic approaches to improve access control in the cloud services [4], [5]. Cryptographic primitives have been proposed for enabling access control on encrypted storage, such as broadcast encryption [6], proxy re-encryption [7], role-based encryption [8] and attribute-based encryption [1]. For the reason of security, scalability and flexibility, ABE has been regarded as one of the most suitable technologies for enabling access control [9]. Users whose attributes satisfying the access policy are able to access the plain data. ABE is mainly classified into two complementary forms, key-policy ABE [10] and ciphertext-policy ABE [2], [3]. In CP-ABE, attributes are used to describe the user's attributes and access policies over these attributes are attached to the encrypted data. Due to its flexibility and expressiveness, CP-ABE has more applications in cloud storage access control [11], [12], [13]. In this paper, we borrow CP-ABE as a component into our SACS design.

Sanitizable Signatures : Sanitizable signatures (SS's) are proposed by Ateniese et al. [14] to allow controlling modifications of signed messages without invalidating the signature. SS is a variant of digital signatures where a designated party (the sanitizer) can update admissible parts of a signed message. Brzuska et al. [15] introduced most of security notions in SS's. Fehr and Fischlin [16] proposed sanitizable signcryption to hide the message-signature pair from the sanitizer. Many SS schemes [17], [18], [19], [20] have been proposed to satisfy different properties. SS provides the foundation to the concept of sanitization in encryption.

Access Control Encryption : Access Control Encryption (ACE) [21] was introduced to provide fine-grained access control. ACE gives different rights to different users not only in terms of which messages they are allowed to receive, but also which messages they are allowed to send. Here, the important property of Sanitization is included. ACE can prevent corrupted senders from sending information to corrupted receivers. In ACE, the sanitizer uses its sanitizer key from the authority to execute a specific randomized algorithm on the incoming ciphertext and thereafter passes the result to a database server or the receivers. By sanitizing, ACE ensures that no matter what the corrupted sender sends, what the receiver receives looks like a random encryption of a random message. In our SACS, the sanitizing operation does not need a sanitizer key from the authority. Only the valid receiver, who is assigned a valid private key by the authority, can recover the message.

3. METHODOLOGY

3.1EXISTING SYSTEM

In Existing, Access Control Encryption (ACE) was introduced to provide fine-grained access control. ACE gives different rights to different users not only in terms of which messages they are allowed to receive, but also which messages they are allowed to send. Here, the important property of Sanitization is included. ACE can prevent corrupted senders from sending information to corrupted receivers. In ACE, the sanitizer uses its sanitizer key from the authority to execute a specific randomized algorithm on the incoming ciphertext and thereafter passes the result to a database server or the receivers.

By sanitizing, ACE ensures that no matter what the corrupted sender sends, what the receiver receives looks like a random encryption of a random message. In our approach, the sanitizing operation does not need a sanitizer key from the authority. Only the valid receiver, who is assigned a valid private key by the authority, can recover the message.

3.1.1DISADVANTAGES OF EXISTING SYSTEM

- Inflexible.
- Not able to completely restrict Malicious Data Access.
- Inefficient.

3.2 PROPOSED METHODOLOGY

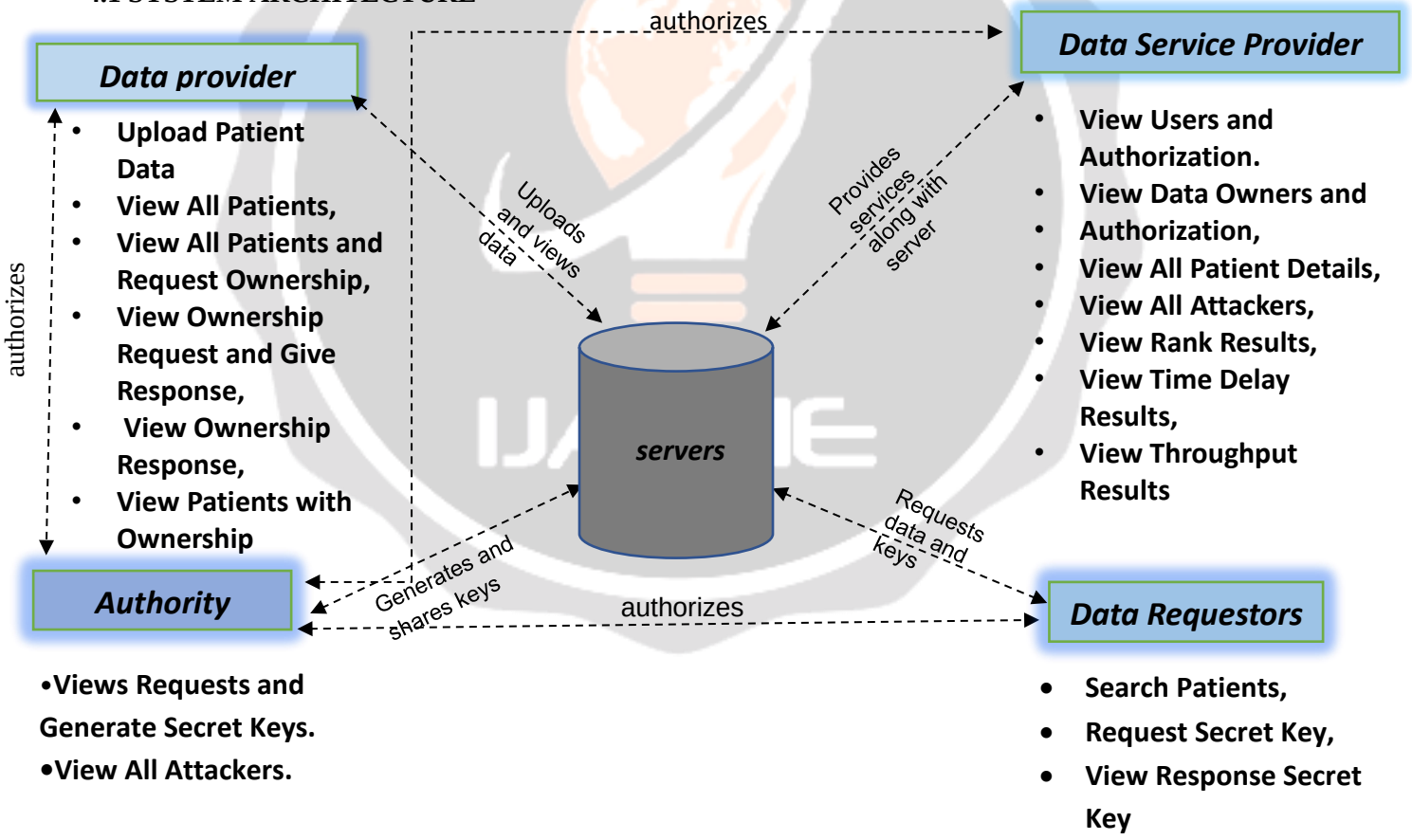
In Proposed System, aim to propose a very practical notion, called Sanitizable Access Control System, or simply SACS, which is designed for the cloud storage to resist against malicious data publishers. SACS enables a flexible access control for both data publishers and data receivers.

As in ABE, SACS allows any valid receivers who are equipped with private keys satisfying the access policy to decrypt the ciphertext. However, SACS is equipped with sanitizing capability, which prevents malicious data publishers from generating ciphertexts that will be decryptable without any valid private keys. Although the malicious data publishers can maliciously generate ciphertexts which can be decrypted by anyone, the sanitizer will transform these ciphertexts into new ciphertexts which will only be decryptable by valid private key holders and present our architecture as well as our scheme to achieve the above concept to build SACS.

4. SYSTEM DESIGN

It is a process of planning a new business system or replacing an existing system by defining its components or modules to satisfy the specific requirements. Before planning, you need to understand the old system thoroughly and determine how computers can best be used in order to operate efficiently.

4.1 SYSTEM ARCHITECTURE



4.2 MODULES

In this Proposed System, There are Four Modules. They are:

1) Data Provider.
2) DSP.

3) Authority.
4) Data Requestors.

4.2.1 DATA PROVIDER

In this module, Data Providers has to register to cloud and logs in, Encrypts and uploads a file to cloud server and also performs the following operations such as Upload Patient Data, View All Patients, View All Patients and Request Ownership, View Ownership Request and Give Response, View Ownership Response, View Patients with Ownership.

- Upload Patient Data
- View All Patients,
- View All Patients and Request Ownership,
- View Ownership Request and Give Response,
- View Ownership Response,
- View Patients with Ownership

4.2.2 DATA SERVICE PROVIDER

In this module the cloud will authorize both the owner and the user and also performs the following operations such as View Users and Authorization, View Data Owners and Authorization, View All Patient Details, View All Attackers, View Rank Results, View Time Delay Results, View Throughput Results.

- View Users and Authorization.
- View Data Owners and
- Authorization,
- View All Patient Details,
- View All Attackers,
- View Rank Results,
- View Time Delay Results,
- View Throughput Results

4.2.3 AUTHORITY

In this module, the Authority performs the following operations such as View Request and Generate Secret Key, View All Attackers.

- Views Requests and Generate Secret Keys.
- View All Attackers.

4.2.4 DATA REQUESTORS

In this module, the user has to register to cloud and log in and performs the following operations such as Search Patients, Request Secret Key, View Response Secret Key.

- Search Patients,
- Request Secret Key,
- View Response Secret Key.

5. RESULTS AND PERFORMANCE

EXECUTION PROCEDURE
The Execution procedure is as follows:

The Sanitizable Access Control System for Secure Cloud Storage against Malicious Data Publishers is a scheme designed to protect data stored in the cloud from unauthorized access and tampering, especially from malicious data publishers. The execution procedure for this system typically involves the following steps:
A. System Setup:
The trusted authority (TA) generates and distributes system parameters, including public parameters and master keys.The TA also initializes the sanitizer, which is responsible for sanitizing (modifying or redacting) data before it is stored in the cloud.

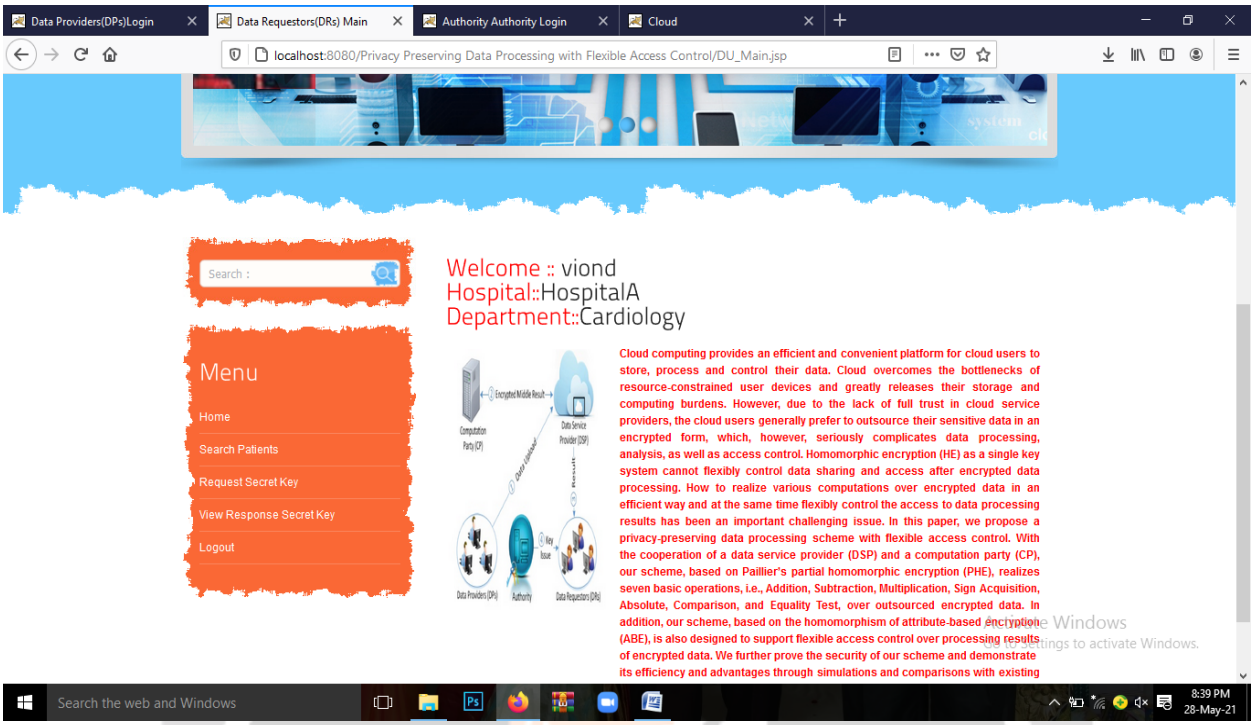
B. User Registration:
Users (data owners and data consumers) register with the TA by providing their identities and other necessary information. The TA issues secret keys and credentials to the registered users.

C. Data Publication:
a. Data Encryption: A data owner encrypts their data using an encryption scheme, such as attribute-based encryption (ABE), and generates an access policy that specifies the attributes required for decryption.
b. Data Sanitization: The data owner sends the encrypted data and access policy to the sanitizer. The sanitizer applies sanitization techniques to the encrypted data and access policy, removing or modifying sensitive information based on predefined rules or policies.
c. Data Storage: The sanitized encrypted data and modified access policy are uploaded to the cloud storage by the sanitizer.

D. Data Access:
a. User Authentication: A data consumer (authorized user) sends an access request to the TA, along with their attributes or credentials. The TA verifies the user's attributes and credentials and issues a decryption key if the user satisfies the access policy.
b. Data Retrieval: The authorized user retrieves the sanitized encrypted data and modified access policy from the cloud storage.
c. Data Decryption: Using the decryption key provided by the TA, the authorized user decrypts the sanitized encrypted data to obtain the original data in a sanitized form (with sensitive information removed or modified).

E. Key Update and Revocation:
The system may include mechanisms for periodically updating encryption keys, access policies, or revocation lists to maintain the security and access control of the stored data.
The Sanitizable Access Control System for Secure Cloud Storage against Malicious Data Publishers incorporates several security measures to protect data confidentiality, integrity, and access control.


WhatsApp
Owner Main
Server Main
HOME Page
View Users
New Tab
localhost:8080/Privacy Preserving Data Processing with Flexible Access Control/
Search :
Menu
Data Requestors(DRs)
Data Providers(DPs)
Cloud Server
Authority
Introduction
Cloud computing provides an efficient and convenient platform for cloud users to store, process and control their data. Cloud overcomes the bottlenecks of resource-constrained user devices and greatly releases their storage and computing burdens. However, due to the lack of full trust in cloud service providers, the cloud users generally prefer to outsource their sensitive data in an encrypted form, which, however, seriously complicates data processing, analysis, as well as access control. Homomorphic encryption (HE) as a single key system cannot flexibly control data sharing and access after encrypted data processing. How to realize various computations over encrypted data in an efficient way and at the same time flexibly control the access to data processing results has been an important challenging issue. In this paper, we propose a privacy-preserving data processing scheme with flexible access control. With the cooperation of a data service provider (DSP) and a computation party (CP), our scheme, based on Paillier's partial homomorphic encryption (PHE), realizes seven basic operations, i.e., Addition, Subtraction, Multiplication, Sign Acquisition, Absolute, Comparison, and Equality Test, over outsourced encrypted data. In addition, our scheme, based on the homomorphism of attribute-based encryption (ABE), is also designed to support flexible access control over processing results of encrypted data. We further prove the security of our scheme and demonstrate its efficiency and advantages through simulations and comparisons with existing work..
Activate Windows
Go to Settings to activate Windows.
Search the web and Windows
4:29 PM
28-May-21


Data Providers(DPs)Login
Data Requestors(DRs) Main
Authority Authority Login
Cloud
localhost:8080/Privacy Preserving Data Processing with Flexible Access Control/DU_Main.jsp
Search :
Menu
Home
Search Patients
Request Secret Key
View Response Secret Key
Logout
Welcome :: viond
Hospital::HospitalA
Department::Cardiology
Cloud computing provides an efficient and convenient platform for cloud users to store, process and control their data. Cloud overcomes the bottlenecks of resource-constrained user devices and greatly releases their storage and computing burdens. However, due to the lack of full trust in cloud service providers, the cloud users generally prefer to outsource their sensitive data in an encrypted form, which, however, seriously complicates data processing, analysis, as well as access control. Homomorphic encryption (HE) as a single key system cannot flexibly control data sharing and access after encrypted data processing. How to realize various computations over encrypted data in an efficient way and at the same time flexibly control the access to data processing results has been an important challenging issue. In this paper, we propose a privacy-preserving data processing scheme with flexible access control. With the cooperation of a data service provider (DSP) and a computation party (CP), our scheme, based on Paillier's partial homomorphic encryption (PHE), realizes seven basic operations, i.e., Addition, Subtraction, Multiplication, Sign Acquisition, Absolute, Comparison, and Equality Test, over outsourced encrypted data. In addition, our scheme, based on the homomorphism of attribute-based encryption (ABE), is also designed to support flexible access control over processing results of encrypted data. We further prove the security of our scheme and demonstrate its efficiency and advantages through simulations and comparisons with existing
Activate Windows
Go to Settings to activate Windows.
Search the web and Windows
8:39 PM
28-May-21


Data Providers(DPs)Main
Data Requestors(DRs) Main
Authority Main
Cloud
localhost:8080/Privacy Preserving Data Processing with Flexible Access Control/DO_Main.jsp
Search :
Menu
Home
Upload Patient Data
View All Patients
View All Patients and Request Ownership
View Ownership Request and Give Response
View Ownership Response
View Patients with Ownership
Logout
Welcome bhanu
Cloud computing provides an efficient and convenient platform for cloud users to store, process and control their data. Cloud overcomes the bottlenecks of resource-constrained user devices and greatly releases their storage and computing burdens. However, due to the lack of full trust in cloud service providers, the cloud users generally prefer to outsource their sensitive data in an encrypted form, which, however, seriously complicates data processing, analysis, as well as access control. Homomorphic encryption (HE) as a single key system cannot flexibly control data sharing and access after encrypted data processing. How to realize various computations over encrypted data in an efficient way and at the same time flexibly control the access to data processing results has been an important challenging issue. In this paper, we propose a privacy-preserving data processing scheme with flexible access control. With the cooperation of a data service provider (DSP) and a computation party (CP), our scheme, based on Paillier's partial homomorphic encryption (PHE), realizes seven basic operations, i.e., Addition, Subtraction, Multiplication, Sign Acquisition, Absolute, Comparison, and Equality Test, over outsourced encrypted data. In addition, our scheme, based on the homomorphism of attribute-based encryption (ABE), is also designed to support flexible access control over processing results of encrypted data. We further prove the security of our scheme and demonstrate its efficiency and advantages through simulations and comparisons with existing work..
Activate Windows
Go to Settings to activate Windows.
Search the web and Windows
9:03 PM
28-May-21

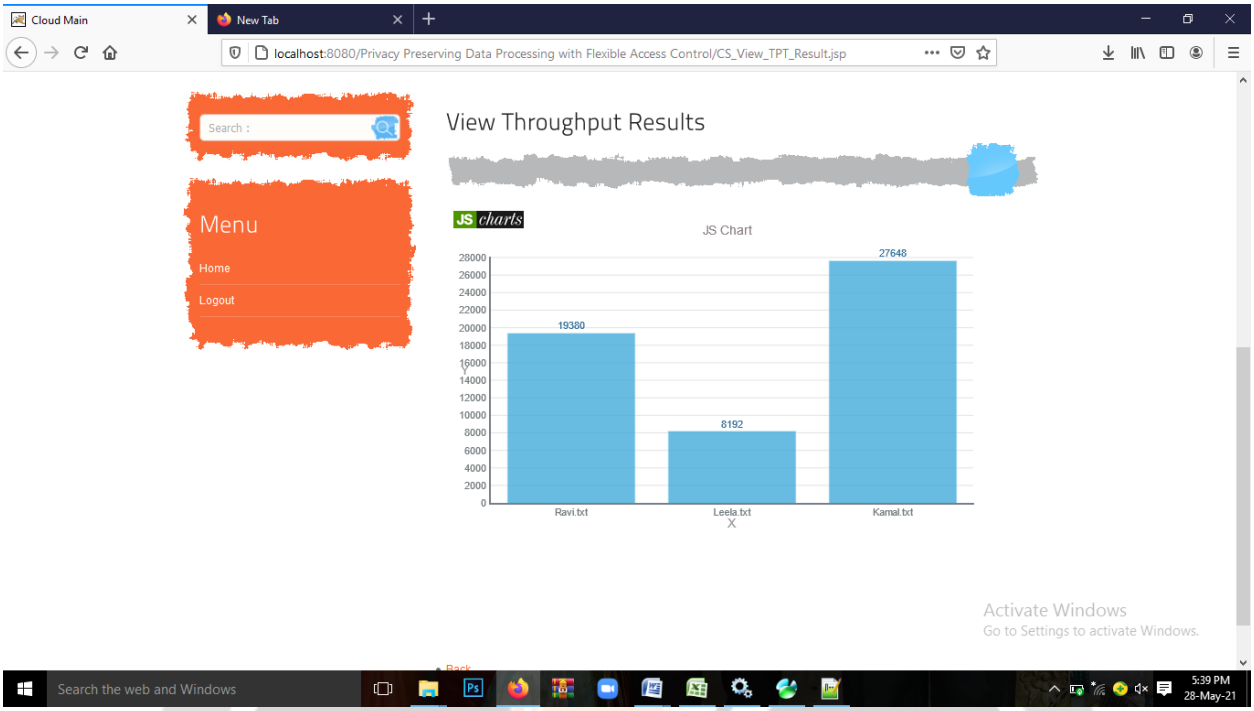


Fig. Datasets Trained and Tested Results

6. 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 actually be decrypted by unauthorized users without the need of valid keys. We designed a system and its secure scheme to enable protection against this kind of attack. Our scheme is proven secure under q-Parallel Bilinear Diffie-Hellman Exponent Assumption. We also provided an implementation of our system for performance analysis.

6.1 FUTURE SCOPE

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.

.

7. REFERENCE

[1] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in Proceedings of the 13th ACM Conference on Computer and Communications Security, CCS 2006, 2006, pp. 89–98.

[2] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” in 2007 IEEE Symposium on Security and Privacy (S&P 2007), 2007, pp. 321–334.

[3] B. Waters, “Ciphertext-policy attribute-based encryption: An expressive, efficient, and provably secure realization,” in PKC 2011, ser. LNCS, vol. 6571, 2011, pp. 53–70.

[4] S. Berger, S. Garion, Y. Moatti, D. Naor, D. Pendarakis, A. ShulmanPeleg, J. R. Rao, E. Valdez, and Y. Weinsberg, "Security intelligence for cloud management infrastructures," IBM Journal of Research and Development, vol. 60, no. 4, pp. 11:1–11:13, 2016.

[5] "Secure access control for cloud storage," https://www.research.ibm.com/haifa/projects/storage/cloudstorage/secure access.shtml.

[6] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast encryption with short ciphertexts and private keys," in CRYPTO 2005, ser. LNCS, vol. 3621, 2005, pp. 258–275.

[7] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy re-encryption schemes with applications to secure distributed storage," ACM Trans. Inf. Syst. Secur., vol. 9, no. 1, pp. 1– 30, 2006.