International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 07 | July 2021 www.irjet.net p-ISSN: 2395-0072

DESIGN AND IMPLEMENTATION OF CONNECTED DATALAKE SYSTEM FOR RELIABLE DATA TRANSMISSION

M.Dilli Babu[1], Sathiya Murthi.M[2], Sravan Kumar.P.M[3], Surendran.S[4], Vimalan.G[5]

[1]Professor, Dept. of Information Technology, Panimalar Engineering College, Tamil Nadu, Chennai.
[2-5]Student, Dept. of Information Technology, Panimalar Engineering College, Tamil Nadu, Chennai.

Abstract – *In this project, we design and implement a connected datalake system based on distributed cloud storage. In addition, the system performs real-time error recovery such that the transferred data can be restored to the abnormal point when an abnormality occurs during transmission. The possibility of constructing a secure storage for dynamic data by leveraging the algorithms involved in secure network coding. We show that some of the secure network coding schemes can be used to construct efficient secure protocols for dynamic data.*

1.INTRODUCTION

An existing ABE-based approaches can be further divided into key-policy ABE (KP-ABE) and cipher text by the data owner, and a private key is associated with the attributes of a data user. Such a scheme is particularly suitable for cloud-related applications, For example, the access policy can be specified by enterprise or individuals, and used to encrypt the outsourced data based on the attributes owned by potential users. Only users whose attributes meet the access policy of cipher text can decrypt and obtain the outsourced data successfully. There have been a number of extensions to the basic CPABE schemes.

1.1 Need of the Project

Lewko and Waters considered the property of a large universe and divided ABE into ABE for the small universe (SU-ABE) and ABE for the large universe (LU-ABE). In SU-ABE, the attribute universe of the system is required to be determined when the public parameters are established and no additional attribute can be added, while in LU-ABE, the available attributes are "infinite" (super polynomial size) and new attributes can be added any time. Disadvantage of this existing projects are The performance of this encrypted format is low, It consuming more time and cost. The security and the accuracy is less. Data lake is all about storing large amount of data, which can be structured,semi structured or unstructured. Various types of data can be stored, retrived and modified. The file format can be any type, such as JPEG, JPG, TXT, DOC, PDF, PPT etc. We proposed an efficient large-universe CPABE scheme with public traceability for Datalake storage deployment. We showed that LU-CPABE-PT is key abuse and key escrow resilience. Specifically, LU-CPABEPT allows two authorities KGC and AA to collaboratively generate the private keys of system users. Since neither KGC nor AA knows the final private keys of users and is not able to forge a valid private key, the KGC or AA is not capable of illegally distributing the user's private keys to unauthorized users or decrypting the user's ciphertexts without the user's authorization. We also showed that LU-CPABE-PT supports public traceability (i.e., anyone can trace the owner of abused keys), which in turn implies user key abuse resilience. In addition, we evaluated the performance and security of LU-CPABEPT.

1.2 Objective of the Project

Data lakes are being accepted by business organizations that want to modernize their data platforms. A Data lake is a storage repository that can store a large amount of structured, semi-structured, and unstructured data. It is a place to store every type of data in its native format with no fixed limits on account size or file. It offers high data quantity to increase analytic performance and native integration. A data lake is a large storage repository that holds a vast amount of raw data in its native format until it is needed. An enterprise data lake (EDL) is simply a data lake for enterprise-wide information storage and sharing. A data lake has structured data, unstructured data, machine to machine, logs flowing through in real-time as shown in Fig.1.1

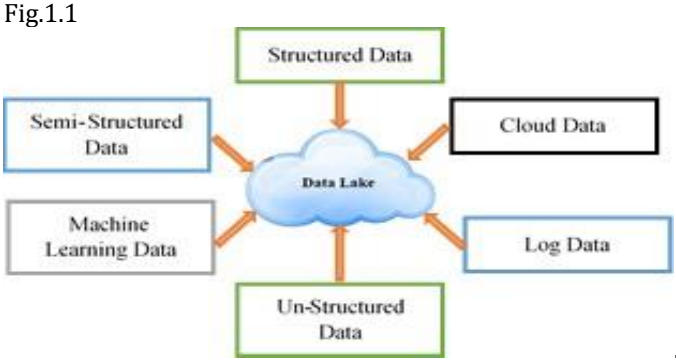

Fig.1 Types of Data in Datalake.

2. SYSTEM DESIGN

2.1 Implementation Methodology

In this method can be implemented in all other websites, for example who wants more security from the attackers. We use this in college, government, company's etc.

2.2 Module Description

In this Project, we have 4 modules. They are as follows:

A. Data User
B. TPA
C. KGC
D. Server

A. Data User

- Register their own details.
- Authorized the AA After login, Login their correct user name and password.
- Upload the files to the cloud in encrypted format with file private and trapdoor key by using fuzzy logic for key generation for encryption & decryption.
- Manage the file.
- Search Files: User can search file with Encrypted format, then sends the request to the key Generator Center.
- View Request Status Waiting or Accept.
- Download Files by using the file private key, user can download the files in decrypted format.
- Logout Login.
- View all Upload file & Attacker them
- Logout.

B. TPA

- Login their correct user name and password.
- View all Data User and Authorized them.
- View user request to Check the Integrity Auditing.
- Logout.

C. KGC

- Login their correct user name and password(KGC).
- View all upload File.
- View all Data User Request details then, send the Private key user mail id.
- Logout.

D. Server

- View all registered Data Users.
- View all Searchable History details.
- View all Integrity check Request details.
- Result- Generate the result based on the file request counts.
- Number of Download file:- calculate the number of download file to the cloud.
- Logout.

2.3 System Architecture

A key/data encapsulation mechanism is used to construct our cloud storage system in which our newly proposed LUCPABE-PT scheme is applied to encrypt a symmetric session key and then, the key is further used to encrypt the outsourced data. The cloud storage system mainly contains five entities, namely, KGC, AA, data owner (DO), data user (DU), and cloud server (CS).

1) KGC is considered to be a semi trusted entity which participates in the generation of system public parameters and user's secret keys. For fine-grained access control, the user's attributes are embedded in the user's secret keys. In addition, KGC inserts the hash value of the identity of the user into the user's secret key for public traceability.

2) AA is also considered to be a semi trusted entity. Similarly, AA participates in the generation of system public parameters and user's private keys. However, neither AA nor KGC knows the whole user's private keys.

3) DO is a data owner who specifies an access control policy and a symmetric session key before outsourcing the data. By using the KEM/DEM mechanism, the DO encrypts the outsourced data and uploads it to the CS.

4) DU is a data user who is determined by a unique identity id and a group of attributes S. DU owns a private key associated with id and S for authorized decryption.

5) CS has sufficient storage and computation capabilities and is responsible for storing all outsourced KEM/DEM ciphertexts.

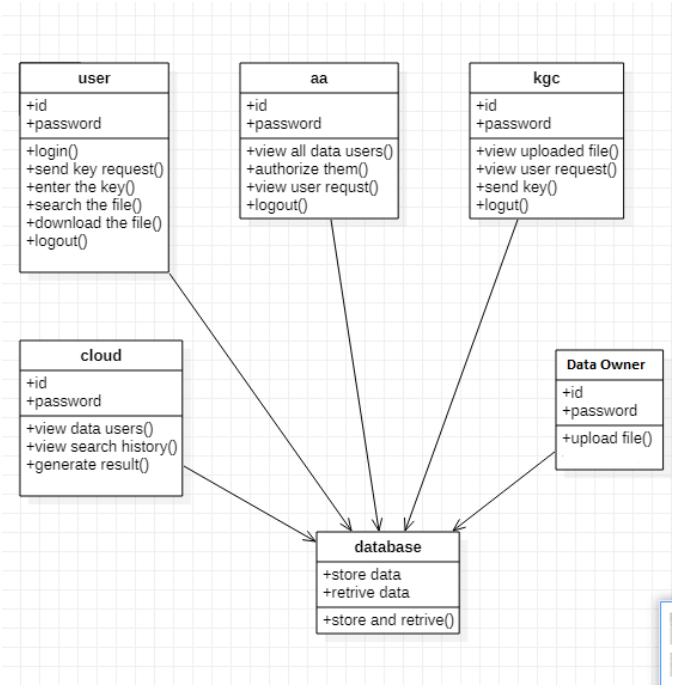

Fig.2. Class Diagram for the Proposed System.

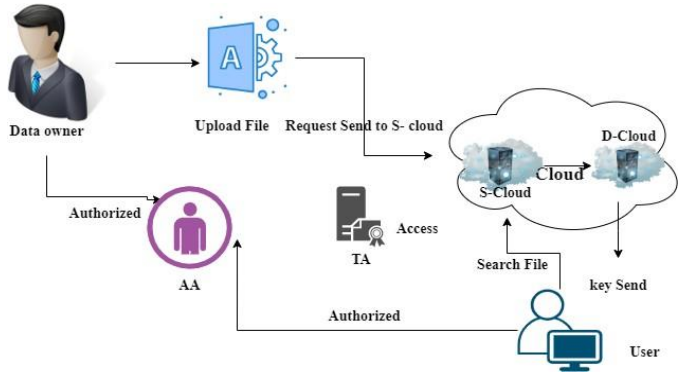

Fig.3 System Architecture Diagram.

3. FEASIBILITY STUDY

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

4. REQUIREMENT SPECIFICATION

4.1 Requirement Analysis and Specifications

The requirement engineering process consists of feasibility study, requirements elicitation and analysis, requirements specification, requirements validation and requirements management. Requirements elicitation and analysis is an iterative process that can be represented as a spiral of activities, namely requirements discovery, requirements classification and organization, requirements negotiation and requirements documentation.

4.2 Input Requirements

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

i. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
ii. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
iii. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus, the objective of input design is to create an input layout that is easy to follow.

4.3 Output Requirements

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

i. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.
ii. Select methods for presentinginformation.
iii. Create document, report, or other formats that contain information produced by the system.
iv. The output form of an information system should accomplish one or more of the following objectives:
    a. Convey information about past activities, current status or projection of Future.
    b. Signal important events, opportunities, problems or warnings.
    c. Trigger an action

d. Confirm an action.

4.4 Functional Requirement

Functional requirements involve calculations, technical details, data manipulation and processing, and other specific functionality that define what a system is supposed to accomplish. The plan for implementing functional requirements is detailed in the system design, whereas non-functional requirements are detailed in the system architecture.

As defined in requirements engineering, functional requirements specify particular results of a system. This should be contrasted with non-functional requirements, which specify overall characteristics such as cost and reliability. Functional requirements drive the application architecture of a system, while non- functional requirements drive the technical architecture of a system.

4.5 Requirement Analysis and Specifications

Software requirements is a sub-field of Software engineering that deals with the elicitation, analysis, specification, and validation of requirements for software. Requirements analysis in systems engineering and software engineering, encompasses those tasks that go into determining the needs or conditions to meet for a new or altered product, taking account of the possibly conflicting requirements of the various stakeholders, such as beneficiaries or users. Requirements analysis is critical to the success of a development project. Requirements must be actionable, measurable, testable, related to identified business needs or opportunities, and defined to a level of detail sufficient for system design.

5. PROTOTYPE IMPLEMENTATION

5.1 Prototype Development

This approach to data analysis in Data Lakes saves a lot of upfront work that usually goes into creating the data structure, thus allowing fast ingestion and storage of data. Moving structuring data to the last step is helpful in situations when the structure itself is hard to define and subject to changes or different interpretations.

5.2 Software Development

Data Lake management deals with the challenges of monitoring and logging the transformations of data as it moves through different layers of the Data Lake. All actions performed on the data are logged, as well as all user actions that led up to them.

6. WORKING

Our scheme any private keys modified by malicious users cannot be successfully used for decryption. In the event that some user illegally shares his/her original private key, the scheme has in place a mechanism to trace the abused private key. Hence, our scheme supports public traceability, key abuse, and key escrow. In addition, our scheme is based on prime order bilinear groups, and is shown to be selectively secure in the standard model.

We construct a large-universe CP-ABE scheme to support public traceability (LU- CPABE-PT) in this article. Our proposed scheme is also designed to mitigate key abuse and key escrow. A summary of LU-CPABE-PT's features is presented as follows:

1) Large Universe: LU-CPABE-PT does not impose a limitation on the capacity of the attribute universe. In addition, the public parameters' size is constant. Unlike SU-ABE where the attribute universe has to be prespecified during the setup stage and no additional attribute can be subsequently added, LU-CPABE- PT allows new attributes to be added to the system. Thus, LU-CPABE-PT is scalable and practical for deployment in the cloud storage environment since the system users (and the corresponding attribute universe) change dynamically.

2) Public Traceability: In LU-CPABE-PT, the individual user's identity is embedded directly into his/her private key in the plaintext form. This facilitates subsequent key abuse investigation. For example, once a key is determined to be abused, the user's identity of the abused key can be determined in a timely manner. Our approach also significantly reduces both calculation and communication overheads for tracking authorities, in comparison to private traceability.

3) Minimal Storage on Tracing: LU-CPABE-PT removes the cost of maintaining an additional identity table for tracing because the user's identity associated with an abused key can be visually observed and easily extracted.

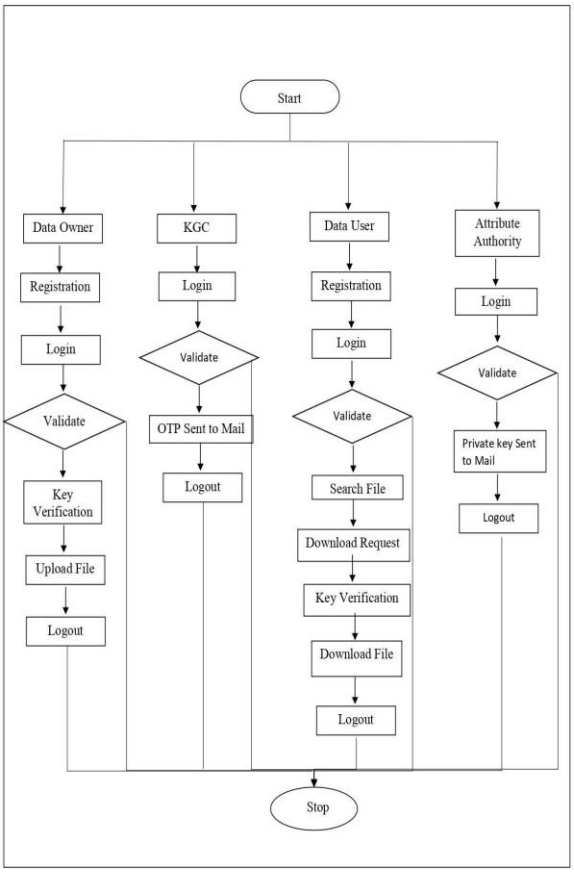

Fig.4. Flow chart diagram for the Datalake System.

4) No Key Escrow: In LU-CPABE-PT, the user's private key is generated collaboratively by two semitrust authorities, namely, a key generation center (KGC) and an AA. Both KGC and AA are not allowed to access the full decryption key and are unable to forge it. This makes it computationally challenging for an individual authority to successfully decrypt ciphertexts.

5) No Key Abuse: If a malicious user reveals his/her original private key, we can use the trace algorithm to trace the offending user. In addition, if a malicious user modifies his/her key to avoid tracking, then the modified key cannot be used for decryption anymore. This implies LU-CPABE-PT is resilient to user key abuse. We also remark that since neither KGC nor AA can obtain a complete decryption key, no individual authority has the ability to redistribute the private key. In other words, LU-CPABE-PT is resilient to authority key abuse.

7. ALGORITHM

1) Ciphertext:

Ciphertext is also known as encrypted or encoded information because it contains a form of the original plaintext that is unreadable by a human or computer without the proper cipher to decrypt it. Decryption, the inverse of encryption, is the process of turning ciphertext into readable plaintext.

2) AES:

The algorithm described by AES is a symmetric-key algorithm, meaning the same key is used for both encrypting and decrypting the data. The Advanced Encryption Standard (AES) is a symmetric block cipher chosen by the U.S. government to protect classified information. It is essential for government computer security, cybersecurity and electronic data protection. Symmetric, also known as secret key, ciphers use the same key for encrypting and decrypting, so the sender and the receiver must both know -- and use -- the same secret key. The government classifies information in three categories: Confidential, Secret or Top Secret. All key lengths can be used to protect the Confidential and Secret level. Top Secret information requires either 192- or 256-bit key lengths.

8. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

No	Test Scenario	Expected Result	Test Result
1	Username is correct. Password is incorrect.	Username and Password is incorrect.	Username and Password is incorrect.
2	Username is incorrect. Password is correct.	Username and Password is incorrect.	Username and Password is incorrect.
3	Username is empty. Password is correct.	Username is required.	Username is required.
4	Username is correct. Password is empty.	Password is required.	Password is required
5	Both Username and Password is incorrect.	Username and Password is incorrect.	Username and Password is incorrect.
6	Both Username and Password is empty.	Username and Password is required.	Username and Password is required.
7	Both Username and Password is correct.	Login Successful.	Login Successful.

Fig.5 Testing Table in various scenarios.

9. ADVANTAGES

- High security and more effective.
- This system gives more accuracy compared to another systems.
- No illegal key share doesn't happen, because it should be traced.
- It supports public traceability, key abuse, and key escrow.

10. FUTURE ENHANCEMENTS AND CONCLUSION

10.1 Future Enhancements

We remark that there is also the concept of "revocation" in ABE, which comprises "user revocation" and "attribute revocation." It is generally believed that attribute revocation is a more fine-grained revocation than user revocation, but its implementation process is more complicated. How to achieve an efficient ABE scheme with public traceability and attribute revocation simultaneously is a direction for our future research.

10.2 Conclusion

We proposed an efficient large-universe CPABE scheme with public traceability (LU-CPABE-PT) for cloud storage deployment. We showed that LU- CPABE-PT is key abuse and key escrow resilience. Specifically, LU-CPABEPT allows two authorities KGC and AA to collaboratively generate the private keys of system users. Since neither KGC nor AA knows the final private keys of users and is not able to forge a valid private key, the KGC or AA is not capable of illegally distributing the user's private keys to unauthorized users or decrypting the user's ciphertexts without the user's authorization. We also showed that LU- CPABE-PT supports public traceability (i.e., anyone can trace the owner of abused keys), which in turn implies user key abuse resilience. In addition, we evaluated the performance and security of LU-CPABEPT. Specifically, we proved that LU-CPABE-PT is selectively secure under the decisional q-parallel BDHE assumption in the standard mode

REFERENCES

[1] X. Fu, X. Nie, T. Wu, and F. Li, "Large universe attribute-based access control with efficient decryption in cloud storage system," J. Syst. Softw., vol. 135, pp. 157–164, Jan. 2018. [11] M. Chase, "Multi-authority attribute-based encryption," in Proc. Theory Cryptography Conf., 2007, pp. 515–534.

[2] A. Lewko and B. Waters, "Decentralizing attribute-based encryption," in Proc. Annu. Int. Conf. Theory Appl. Cryptograph. Techn., 2011, pp. 568–588.

[3] Z. Liu, Z. Cao, Q. Huang, D. S. Wong, and T. H. Yuen, "Fully secure multi- authority ciphertext-policy attribute-based encryption without random oracles," in Proc. Eur. Symp. Res. Comput. Security, 2011, pp. 278–297.

[4] G. Yu, X. Ma, Z. Cao, W. Zhu, and J. Zeng, "Accountable multi-authority ciphertext-policy attribute-based encryption without key escrow and key abuse," in Proc. Int. Symp. Cybersp. Safety Security, 2017, pp. 337–351.

[5] H. Zhong, W. Zhu, Y. Xu, and J. Cui, "Multi-authority attribute-based encryption access control scheme with policy hidden for cloud storage," Soft Comput., vol. 22, no. 1, pp. 243–251, 2018.