Data integrity and revocable attribute-based encryption in clouds

C.Anuradha
Department of Computer Science and Engineering
SCAD College of Engineering and Technology
TamilNadu, India

A.Merlin Ponselvi
Department of Computer Science and Engineering
SCAD College of Engineering and Technology
TamilNadu, India
merlin.ponselvi@gmail.com

Abstract— **One of the most promising application platforms to address the exponential growth in data sharing is cloud computing. Users must encrypt their data before sharing it on the cloud in order to prevent data leaks. It enables clients with constrained computing resources to outsource their heavy computing tasks to the cloud and profitably take advantage of the enormous computing power, bandwidth, storage, and even relevant software that can be shared in a pay-per-use way. The importance of access control cannot be overstated because it serves as the first line of defense against unauthorized access to shared data. On the one hand, the workloads for outsourced computation frequently include sensitive data, including firm financial records, confidential research information, and individually identifiable health data. Sensitive data must be encrypted prior to outsourcing in order to guarantee end-to-end data confidentiality assurance in the cloud and elsewhere. This will help prevent unauthorized information leaking. Clients run the danger of crucial exposure that goes unnoticed but was evident in earlier studies. The practical application of ABE is further restricted by the significant client decryption overhead. The suggested cooperative technique successfully addresses both the issue of key exposure and key escrow. We provide a specific RABE-DI scheme and demonstrate its integrity and confidentiality inside the specified security architecture. In the meanwhile, it significantly lowers client decryption overhead. However, standard data encryption methods effectively prevent the cloud from carrying out any useful operations on the underlying cipher text-policy, which makes the task of computing over encrypted data very challenging.**

Keywords—access control, data leaks, key exposure, key escrow, cipher text-policy.

I. INTRODUCTION

In order to provide faster innovation, flexible resources, and scale economies, cloud computing is the delivery of computing services via the Internet ("the cloud"), including servers, storage, databases, networking, software, analytics, and intelligence. The majority of the time, you only pay for the cloud services you actually use, which helps you cut operational expenses, manage your infrastructure more effectively, and grow as your company's needs evolve.

The authority in cloud computing accepts the user's enrolment and sets up basic parameters. A cloud service provider (CSP) oversees cloud servers and offers a range of services to customers. The data owner generates the cipher-text, encrypts it, and uploads it to CSP. The interested cipher text is downloaded and decrypted by the user from CSP. Typically, shared files are organised hierarchically. In other words, a file department is split up into a number of hierarchy sub-departments that are located at various access levels. The storage cost of cipher text and the time cost of encryption might be reduced if the files in the same hierarchical structure could be encrypted by an integrated access structure. At the moment, an increasing number of plans use encryption to manage the data in the cloud. It enables clients with constrained computing resources to outsource their heavy computation workloads to the cloud and profit financially from the enormous computing power, data transfer capacity, stockpiling, and even appropriate programming that can be accessed in a compensation for each utilisation way.

A cutting-edge way of thinking about computing, distributed computing provides flexible, on-demand, and labor-saving use of computing resources. Unexpectedly, these points of interest are the causes of security and protection problems, which appear because the information allegedly claimed by various customers is stored on some cloud servers rather than under their own control. Distributed computing still has unresolved security issues. Different strategies based on attribute-based encryption have been used to address security challenges.

According to one point of view, workloads for outsourced computing projects usually include sensitive data, such as firm financial records, difficult research data, or eventually identifiable prosperity data, etc. Sensitive data must be combined before outsourcing in order to provide end-to-end data protection assurance in the cloud and past, preventing unauthorised information leakage. However, standard data encryption practises typically prevent the cloud from carrying out any crucial operations of the crucial figure content game plan, making the number of encoded data a problematic issue. Due to its dynamic structure, the proposed plot does not simply attain flexibility. We provide you with the safety of secure computing in an open, communal setting. In our business, we implement advanced property base security. Cloud specialists, domain experts, and clients are in ascending order.

Cloud specialists can easily add or remove the private cloud specialised domain while keeping all of the other areas of interest in the general cloud. The subject-matter expert might evacuate or make the clients inside the zone. Private clientele are what we call them. Private cloud clients and open cloud clients are the two categories of clients. Clients of private clouds depend on available space, whereas those of public clouds depend on cloud experts. Customers have two options for document transfer: public and private. If a private client transfers a general society document, customers can access that document without any security verification because it is only perceivable and available within the region itself. Any cloud client can access a document if the general population client adds users to general documents since the record is always visible and accessible. Private transfers only If a private client transfers a private document, it suggests that the record's perceivability is constrained to the physical realm, but the emit key (OTP), which denotes who has access to the record, determines the document's openness. Anybody can view a private document when it is transferred to a general society client, suggesting that the document's perceivability is open; However, if they are eligible for an access benefit (OTP), they can easily view the document.

A. Scope of the paper

Achieving scalable, adaptable and fine-grained access control of data that has been outsourced in cloud computing. Sensitive data, including commercial financial records, confidential research data, and individually identifiable health information, is included in the workloads for outsourced computation. Users may attempt to access the data files without authorization. As a result, a hierarchy is suggested in which a certain group of users respects a domain authority. The trustworthy authority is also trusted by the domain authority.

B. Objective of the paper

We ensure secure privacy in public social cloud computing. We use hierarchical attribute-based security in our project. Users, Domain authority, and Cloud authority are the attributes in the hierarchy. Cloud authorities can only keep all the information in the overall cloud and have the ability to add or remove domains (private cloud authorities). Users inside a domain can be added or removed by domain administrators. Private users are what we refer to them as. Private cloud and public cloud users are the two different categories of users. Users of private clouds are reliant on the public users' domain under cloud authority. Users have two upload options: public and private. If a private user uploads a public file, only users within the same domain will be able to see it and access it without any further security authentication. Any cloud user can access a public file that has been uploaded by a public user because the file's visibility and accessibility are always public. If a private user uploads a private file that implies that the file is only visible within the domain and that the person who has the secret key (OTP), or the right to view the file, can only access the file. If a public user uploads a private file, the file's visibility is considered to be public. The file is visible to everyone, but only those with the necessary access rights (OTP) can open it.

II. RELATED WORK

A user should only be permitted access to data in some distributed systems if they have a specific set of credentials or attributes. Currently, using a trusted server to store the data and handle access control is the only way to enforce such regulations. The confidentiality of the data will, however, be jeopardised if any server hosting the data is compromised. J. Bethencourt et al. [1] offer Cipher text-Policy Attribute-Based Encryption, a system for implementing complicated access control on encrypted data. Even if the storage server is unreliable, encrypted data can be kept private by employing our procedures. Furthermore, our techniques are resistant to collusion assaults. In the past, attributes-based While prior encryption systems utilised attributes to describe the encrypted data and incorporated policies into the user's keys, our approach uses attributes to describe the user's credentials and relies on the party encrypting the data to decide who is allowed to decrypt it. Therefore, our techniques are conceptually more similar to established access control techniques like Role-Based Access Control (RBAC). The authors also include performance metrics and a description of how their system is implemented. Despite the fact that there are surely other sorts of systems, the advantages of the algorithm enable our system to provide a novel form of encrypted access control where the user's private keys are kept secret. It is more efficient and reliable to replicate data across multiple locations. The algorithm's drawback is there is a good chance that such reductions won't apply to the simpler (and more effective) techniques authors present here. Further investigation that can give this form of security a more solid theoretical base is strongly encouraged.

Zhang, Q., et al. [2] offer a time and attribute based dual access control and data integrity verifiable system in cloud computing applications as a solution to the aforementioned two issues (DCDV). In order to create an effective access time period and a specified decrypt able time period for the user’s attributes key and encrypted data independently, a hierarchical time tree is first presented in the attribute-based encryption technology. The decryption procedure can only be carried out if the user's attribute set complies with the data owner's access policy and the user's attributes key's effective access time period fully overlaps the decryption time period established by the data owner. In order to address the issue of privacy data leakage brought on by private key leaking, the data is dual controlled with time and attributes in this manner. Second, the data verification tree is created using the inverted index and Merkle hash tree. The issue that the cloud server might delete or modify the data is resolved since the data user can independently verify the accuracy of the cipher text data given by the cloud server without having to decrypt it. Finally, the security and efficiency study demonstrates that our plan is both feasible and secure. Due to the advantages of the method, their system enables a novel form of encrypted access control in which the user's private keys are used to implement the HIBE algorithm in their plan for efficient time management. Data replication offers benefits in terms of reliability and performance across several sites. The algorithm's drawbacks include the fact that when the system is configured, the trusted authority chooses a bilinear group and generates some random integers. There will always be numerous exponentiation operations involved in the creation of random numbers. As a result, System Setup's computational complexity.

OutFS, a user-side encrypted file system with a focus on providing transparent encryption for stored and shared outsourced data, is introduced by Khashan, O. A. [3]. Authors use a hybrid encryption scheme structure for OutFS that is based on symmetric and asymmetric techniques. The key management system is logically constructed. OutFS has been linked with the identity-based encryption scheme (IBE) to provide strong data sharing security. The purpose of OutFS is to protect the integrity of file system data structures and outsourced file data. It can be seen from performance analysis and experimental findings that OutFS is effective. It can read and write outsourced files at an average throughput of 10.5 MB/sec and 8.8 MB/sec, respectively. According to security study, OutFS is quite resistant to assaults like brute-force, eavesdropping, man-in-the-middle, and offline dictionary attacks. Despite the algorithm's benefits, the fundamental benefit of their method is that, unlike the CC MAABE, any user can get secret keys from any subset of the system's TAs. High resilience against brute-force, eavesdropping, man-in-the-middle, offline-dictionary, and collusion attacks, high security, high transparency, minimal computing complexity. The system needs to be extremely scalable in terms of key management complexity as well as communication, processing, and storage. They have significant security, effectiveness, and usability problems, and some of their protection strategies are inadequate for cloud data.

In the single-owner multiple-user paradigm, Zhu, J., et al. [4] introduce GSSE, the first general verifiable SSE method that offers verifiability for any SSE technique and additionally allows data updates. The proof index in GSSE is initially decoupled from SSE in order to facilitate result verification in a broader sense. The proof index is then built by the authors using Merkle Patricia Tree (MPT) and Incremental Hash with support for data updates. We also create a timestamp chain to maintain data freshness across many users. There is a tiny overhead for result verification introduced by GSSE, according to careful study and experimental assessments. If all polynomial-time adversaries only have a little edge in the aforementioned security game, the benefits of the technique, DAC-MACS, are secure against static manipulation of authority. The algorithm's drawbacks include VSSE schemes' extremely restricted application, such as their sole support for static databases, their requirement for specialised SSE structures, or their ability to operate only in the single-user paradigm.

In next-generation wireless networks (NGWNs), including 5G cellular networks, IEEE 802.11ax WiFi networks, and wireless ad hoc sensor networks, localization is a crucial problem (WASNs). The notion of radical centres from analytic geometry is used by Chen, Y. S., et al. [5] to propose a 3-D localization procedure for NGWNs using WASNs as an example. Assume that a node that is unknown can determine how far it is from four or more anchor nodes (reference nodes). By selecting four distance measurements to four anchor nodes, a radical centre is calculated. The radical centre is demonstrated to be able to be handled as an estimate of the unknown node position using analytical formulation. Effective filtering procedures are provided to further enhance and fuse these estimations by filtering out the incorrect estimations since every four distance measurements produce one radical centre (in 3-D space). The last guess regarding the location of the unknown node is the answer after averaging the remaining radical centres. Analytical comparisons were made between the suggested algorithm's location mistakes and those produced by the standard minimal mean square error (MMSE) approach. The new algorithm was demonstrated to surpass the traditional MMSE approach in accuracy and efficiency. Numerous computer simulations were run, and the outcomes supported the suggested location algorithm's superiority to the MMSE technique. The algorithm's advantages are as follows: The new algorithm was demonstrated to surpass the traditional MMSE approach in accuracy and efficiency. The findings reveal that none of the revoked users are able to access any files that they have ever decrypted with their old keys. This demonstrates that the algorithms we suggested are reliable and accurate. Because their keys and certificates are no longer valid in the PKI system, revoked users are unable to utilise their current secret seals (SS) to decode the cypher text.

A plan without the service provider's confidence is proposed by Gupta, S., et al. The data owner alone will be responsible for maintaining the security of the data. It would primarily include a tool that would let the data owner choose which people have access to his or her data, whether those rights should be revoked, and whether to notify others of security breaches. The usage of ranked keyword search in this study also enables users to search their files in an encrypted database, which is an advantage over traditional searching methods. The system has the following merits: cost effectiveness, high dependability, resilience, and scalability (access to your data from anywhere, at any time). The system's shortcomings are there are several security and legal hazards to take into account. Unapproved entry IPR protection, publication of sensitive data communication difficulties and risks to data integrity and transmission

A. *Existing System*

The hierarchical structure of shared files hasn't been investigated in CP-ABE, however shared data files typically include traits of a multilevel hierarchy, especially in the healthcare and military sectors. The portion of the cloud storage system that is encrypted is Cipher text-policy attribute-based. The complete management of the file access control authority, which permits all activities on cloud data the produced private key can be used by the key authority to decode all of the encrypted text without the owner's consent, thus it must be entirely trustworthy. Sensitive data must be encrypted before outsourcing in order to prevent unauthorised information leaking. Based on the authority granted, data is encrypted using role-based encryption.

B. *Problem Statement*

Data on compute outsourcing cannot be secure with the current technology. Sensitive data must be encrypted before being outsourced to prevent unauthorised information leaking. Plaintext data in the cloud cannot be secured using standard data encryption methods. Making the task of computing over encrypted data exceedingly challenging. a complicated system of access control laws. As opposed to conventional public key cryptography, cipher-texts are not encrypted for a single user. Multiple values being assigned to the same attribute.

C. *Proposed System*

We provide social cloud computing security. In this paper, we put users, cloud authority, domain authority, and hierarchical security into practise. Only provinces may be added or deleted by the cloud authority, and all information can be saved in the entire cloud. The domain authority has the power to add or remove users from the domain. Private users are those users.

There will be two different users there. The first uses a private cloud, whereas the second uses a public cloud. Public users are subject to cloud authorities, whereas private users receive responses on the domain. There are two options for users when uploading files: public and private.

If only one file is submitted by a private user, convenience and visibility are only available within the domain. All users will have file access rights if a public user uploads a file.

To improve the efficiency and security of key management in attribute-based cipher text encryption for cloud data sharing systems.

File accessibility is determined by who has the secret key (OTP), which indicates who has the right to read the file, if a file is uploaded to a private user, which limits file visibility to the field only. If a public user uploads a private file, then everyone can see the file since visibility is public. However, only those with a privilege, such as a one-time password, can access it.

III. OVERVIEW OF THE PROPOSED WORK

There are five important modules in our proposed system. They are as follows: Data Ownership, Data Consumption, Domain-Level Security, Attribute-Based Security, and Secret File Access

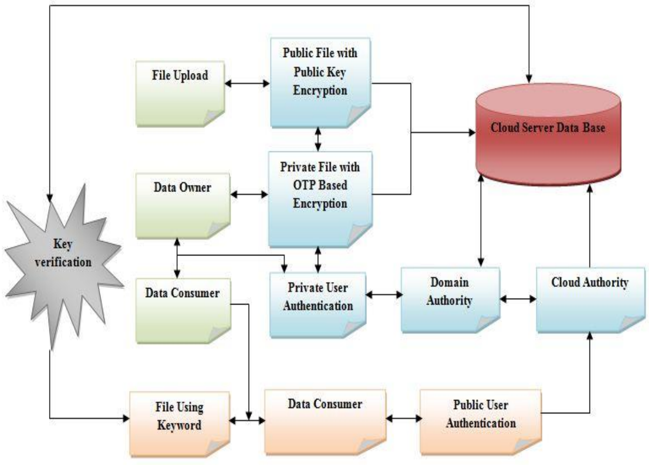


Fig. 1. System Architecture

A. Data Owner

The data owner uploads their data to the cloud server in this module. The data owner encrypts the data file for security reasons before uploading it to the cloud. By altering the expiration date, the data owner can modify the policy governing data files. The owner of the data may be able to alter the encrypted data file. The owner of the data can control who has access to the encrypted data file. Owners of data should assign the majority of the computational burden to cloud servers. KP-ABE is used to smoothly offer fine-grained access control. Each file is encrypted using a symmetric data encryption key that is itself encrypted by a public key that is produced using an access structure and corresponds to a set of KP-ABE properties. It stores the encrypted data file. The related characteristics of a file saved in the cloud must fulfil the access structure of a user's key in order for the user to be able to decrypt the key, which is then used to decrypt the file. To share with data consumers, data owners encrypt and store their data files on the cloud. Data consumers download and decrypt the encrypted data files they want from the cloud in order to access the shared data files. A domain authority manages each data owner consumer. A domain authority is controlled by a trusted authority or its parent domain authority. A hierarchical organisation is used to group data owners, data consumers, domain authorities, and trusted authorities.

B. Data Consumer

Only if the user has access rights to the file may they use the encryption key to open the data file. All rights are granted by the domain authority to users at the user level, and the domain authority alone has control over users' access to data. Users may attempt to access data files inside or beyond the parameters of their access rights, and malevolent users may work together to get sensitive files outside the parameters of their access privileges. To share with data consumers, data owners encrypt and store their data files on the cloud. Data consumers download and decrypt the encrypted data files they want from the cloud in order to access the shared data files. A domain authority oversees the administration of each data owner or consumer. A domain authority is controlled by a trusted authority or its parent domain authority. A hierarchical organisation is used to group data owners, data consumers, domain authorities, and trusted authorities. Online users will constantly be consuming data. While the cloud service provider, the trusted authority, and domain authorities are constantly online, they only go online when it is absolutely essential. It is thought that the cloud has a lot of storage space and processing power. Additionally, we consider data consumers to have read-only access to data files.

To access the cloud storage information and the data consumer entry level based on the hierarchical structure, data consumers must first register an account and then login.

C. Security at the Domain Level

The trusted authority authorizes the top-level domain authorities and serves as the foundation of trust. A domain authority may attempt to get the private keys of users outside of its domain despite the fact that it’s subordinate domain authorities or the users it administers trust it. Users may attempt to access data files inside or beyond the parameters of their access rights, and malevolent users may work together to get sensitive files outside the parameters of their access privileges. We presume that all parties' communication channels are protected by industry-accepted security procedures.

The trusted authority or its parent domain authority is in charge of the domain authority. A hierarchical organisation is used to group data owners, data consumers, domain authorities, and trusted authorities. A federated enterprise is an example of a top-level organisation, while an associated firm in a federated enterprise is an example of a lower-level organisation. Each top-level domain authority corresponds to a top-level organisation. Employees at a company might be data owners or consumers. Each domain authority is in charge of overseeing either the data owners or consumers inside its domain or the domain authorities at the next level. A domain authority may attempt to get the private keys of users outside of its domain despite the fact that it's subordinate domain authorities or the users it administers trust it.

Users may attempt to access data files inside or beyond the parameters of their access rights, and malevolent users may work together to get sensitive files outside the parameters of their access privileges. A trusted authority, various domain authorities, and a large number of users representing data owners and consumers make up the system model. The trusted authority is in charge of creating and distributing root master keys, system settings, and approving top-level domain authorities. A domain authority is in charge of assigning keys to users inside its domain or subordinate domain authorities at the next level. A key structure that details the characteristics of the user's decryption key is given to each user in the system.

D. Attribute-based securit

By adding a delegation mechanism to ASBE, the HASBE method easily accommodates a hierarchical structure of system users. Due to the adaptable attribute set combinations, HASBE not only permits compound attributes but also provides effective user revocation thanks to the various value assignments of the attributes. Based on the security of CP-ABE, we formally established the HASBE's security. a system for access control in cloud computing using hierarchical attribute-set-based encryption (HASBE). In order to achieve scalable, flexible, and fine-grained access control, HASBE augments the cypher text-policy attribute-set-based encryption (CP-ASBE, or ASBE for short) scheme with a hierarchical structure of system users.

E. Access to secret file

To offer a data storage service, the cloud service provider operates a cloud. To share with data consumers, data owners encrypt and store their data files on the cloud. Data consumers download and decrypt the encrypted data files they want from the cloud in order to access the shared data files. The cloud server provider is trustworthy in that it could work with malevolent users (also known as data owners or consumers) to extract files from the cloud and utilise them for their own gain. Each party is assigned a public key and a private key in the system's hierarchical structure, with the latter being retained privately by the party. Users may attempt to access data files inside or beyond the parameters of their access rights, and malevolent users may work together to get sensitive files outside the parameters of their access privileges. By storing encrypted data on servers and keeping the decryption keys confidential, sensitive data is traditionally protected when it is outsourced to other parties.

IV. RESULT AND ANALYSIS

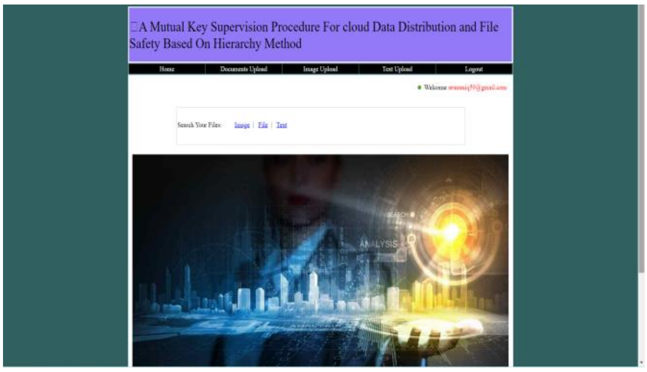
A. Experimental SetUp

This implementation of the proposed method has been carried out on a PC with a Pentium Dual Core 2.3 GHz CPU, hard disk of 250 GB or higher, 2GB RAM, and the Windows operating system. Languages used are Java (JSP, Servlet), HTML. Tools used are JDK 1.7, Net Beans 7.0.1, SQLyog. MySQL is used for Backend.

B. Screenshots

Home Page - In this methodology, the home page is the index form for every user who utilises it as per their requirements. It consists of User Register, User Login, and Admin Login. All the user activities begin with this module. The behaviours of each module is described below.

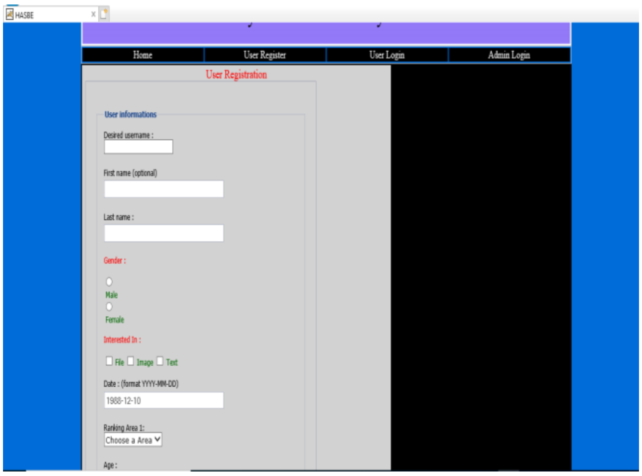
Admin page - Admin is the only module that has access all over the system at any point of time. Whereas the user's belongings may be accessed by the admin module to check what type of file that the user may upload.



A Mutual Key Supervision Procedure For cloud Data Distribution and File Safety Based On Hierarchy Method

Fig. 2. Admin Page

Register Page - For every new user, they need to register their details in the system applications. User registration includes the following information: desired username, firstname, last name, gender, interest in, date, ranking area 1, age, and so on.



User Registration

User informations

Desired username :

First name (optional)

Last name :

Gender :

○ Male

○ Female

Interested In :

□ File □ Image □ Text

Date : (format YYYY-MM-DD)

1988-12-10

Ranking Area 1:

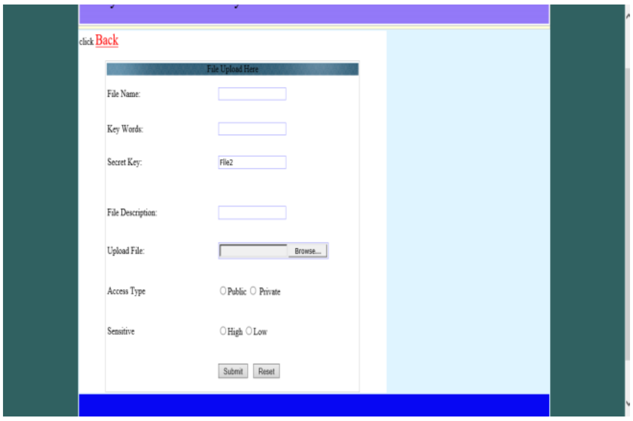
Choose a Area

Age :

Fig. 3. User Registration Page

User Login Page - When a user registers with the system,The user may log into the system by approving the user registration process. The user who registered with valid details may be verified and then accessed to the next module of this system.

File uploading - In this file upload module, the user needs to input his file details along with this upload module. File name, key words, secret key, file description, Upload a file and an access key, which can be either private or public.Sensitivity may vary from low to high.



File Upload Here

File Name:

Key Words:

Secret Key: File2

File Description:

Upload File:

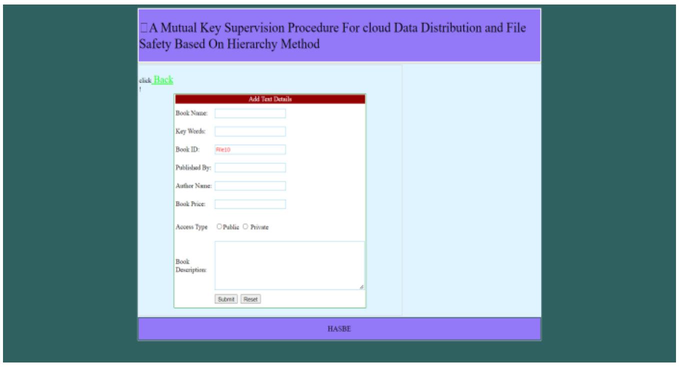
Access Type ○ Public ○ Private

Sensitive ○ High ○ Low

Fig. 4. File Uploading Module

Document type search - This module asks the user to input the file type. The file may be in text file format, image file format, or document file format.

Include Text File Details - The user needs to add the text file input details to get the file encrypted and decrypted. This may happen according to the user inputs.



□A Mutual Key Supervision Procedure For cloud Data Distribution and File Safety Based On Hierarchy Method

Add Text Details

Book Name:

Key Words:

Book ID: File10

Published By:

Author Name:

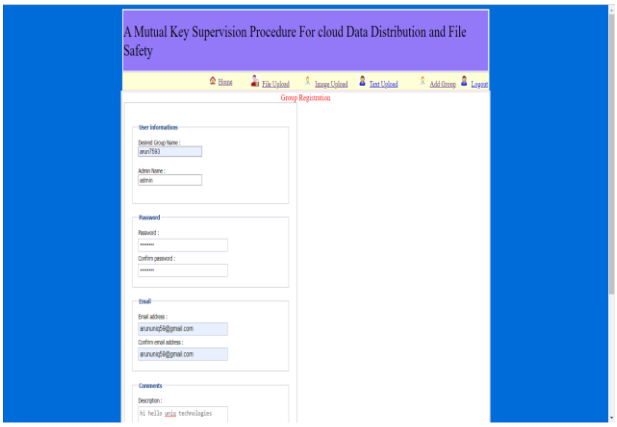
Book Price:

Access Type ○ Public ○ Private

Book Description:

Fig. 5. Adding text file details

User Group Register - A user group can register for this system in which more than two users may be involved with the system to upload the file and get it back in encrypted and decrypted format.



A Mutual Key Supervision Procedure For cloud Data Distribution and File Safety

Group Registration

User informations

Desired Group Name : arun7593

Admin Name : admin

Password

Password : ••••••••

Confirm password : ••••••••

Email

Email address : arununiq59@gmail.com

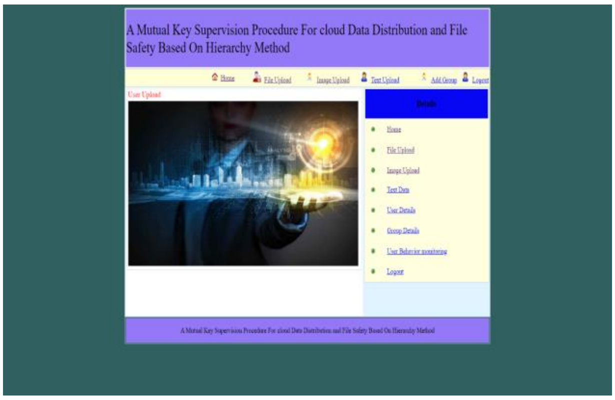
Confirm email address : arununiq59@gmail.com

Comments

Description : hi hello uniq technologies

Fig. 6. User Group Register

User Group Upload - Once the user group has registered in the system, the user may upload their file with this module for group security. This module consists of details like Home, file upload, image upload, text data, user details, group details, user behaviour monitoring, and logout.



A Mutual Key Supervision Procedure For cloud Data Distribution and File Safety Based On Hierarchy Method

A Mutual Key Supervision Procedure For cloud Data Distribution and File Safety Based On Hierarchy Method

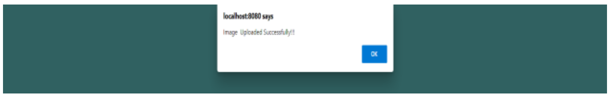
Fig. 7. User Group Upload

C. Outputs

localhost:8080 says

Text uploaded Successfully

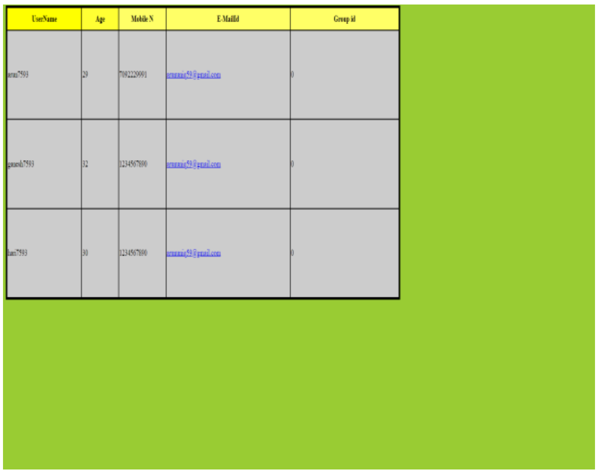
Fig. 8. Successful Upload Of Text File



localhost:8080 says

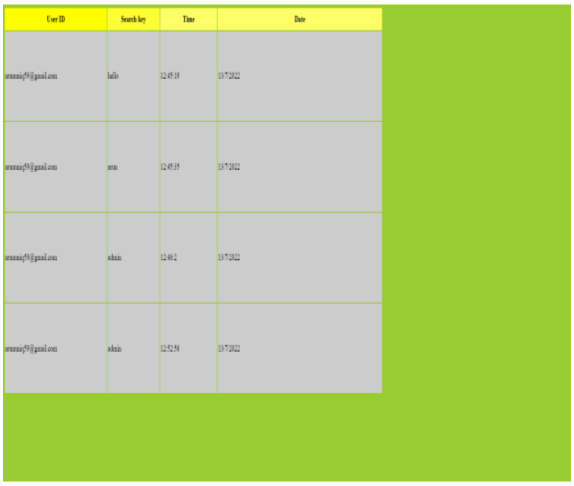
Image Uploaded Successfully!!

Fig. 9. Successful Upload of Iamge File



UserName	Age	Mobile N	E-MailId	Group id
arun7593	29	7092229991	arunraniq59@gmail.com	0
ganesh7593	32	1234567890	arunraniq59@gmail.com	0
hari7593	30	1234567890	arunraniq59@gmail.com	0

Fig. 10. User Details



User ID	Search key	Time	Date
arunraniq59@gmail.com	hello	12:45:19	13/7/2022
arunraniq59@gmail.com	arun	12:45:35	13/7/2022
arunraniq59@gmail.com	admin	12:46:2	13/7/2022
arunraniq59@gmail.com	admin	12:52:56	13/7/2022

Fig. 11. User Behavior

V. CONCLUSION AND FUTURE WORK

Client devices can access data and cloud applications from remote physical servers, databases, and computers over the internet to illustrate how cloud computing functions.. In line with the specified security paradigm, a specific RABE-DI scheme is proposed and its secrecy and integrity are demonstrated. It also significantly lowers the overhead associated with client decryption. But standard data encryption methods effectively restrict the cloud from carrying out any useful operations on the underlying cypher text policy, making computing over encrypted data a highly challenging challenge. Due to its hierarchical nature, the suggested approach also achieves scalability.

This paper will be improved in the future by adopting an automatic trust negotiation unified system for resource preservation. It is possible to create automated trust negotiation using cryptographic credentials.

REFERENCES

[1] Bethencourt, J., Sahai, A., & Waters, B. (2007, May). Ciphertext-policy attribute-based encryption. In 2007 IEEE symposium on security and privacy (SP'07) (pp. 321-334). IEEE.

[2] Zhang, Q., Wang, S., Zhang, D., Wang, J., & Zhang, Y. (2019). Time and attribute based dual access control and data integrity verifiable scheme in cloud computing applications. IEEE Access, 7, 137594-137607.

[3] Khashan, O. A. (2020). Secure outsourcing and sharing of cloud data using a user-side encrypted file system. IEEE Access, 8, 210855-210867.

[4] Zhu, J., Li, Q., Wang, C., Yuan, X., Wang, Q., & Ren, K. (2018). Enabling generic, verifiable, and secure data search in cloud services. IEEE Transactions on Parallel and Distributed Systems, 29(8), 1721-1735.

[5] Chen, Y. S., Deng, D. J., & Teng, C. C. (2016). Range-based localization algorithm for next generation wireless networks using radical centers. IEEE Access, 4, 2139-2153.

[6] Gupta, S., Satapathy, S. R., Mehta, P., & Tripathy, A. (2013, February). A secure and searchable data storage in cloud computing. In 2013 3rd IEEE International Advance Computing Conference (IACC) (pp. 106-109). IEEE.