JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Secure Data De-Duplication Based on Threshold Blind Signature and Bloom Filter in Cloud Storage

[1]K. Spandana,[2]Dharani Dadamoni,[3]Shreya Maramreddy, [4]S. Durgadevi

[1] Assistant Professor, Department of CSE, CBIT, Gandipet,[2] Student Department of CSE, CBIT, Gandipet ,[3]Student Department of CSE, CBIT, Gandipet , Assistant Professor, Department of CSE, CBIT, Gandipet

Abstract

Within the cloud environment, the availability of storage, as well as bandwidth, can be effectively preserved in virtue of data de-duplication. However, refraining redundancy from additional storage or communication is not trivial due to security concerns. Though intensive research has been addressed on a convergent cryptosystem for secure data de-duplication, the conflicts between functionality, confidentiality, and authority remain unbalanced. More concretely, although data are obfuscated under convergent encryption, a violent dictionary attack is still efficacious since the whole pseudorandom process relies heavily on plaintexts. As for data ownership, the download privilege, which depends on hash value, may also be infringed due to the same reason. To dispose of these problems, we presented a conspiracy-free data de-duplication protocol based on a threshold blind signature in this article. With the help of a key server, the outsourced file and de-duplication label will be computationally indistinguishable from random strings. We used the Boom filter as a tool to implement a proof of ownership, ensuring that the ownership claims made by users are real. It effectively prevents the attacker from using the stolen tag to get the whole file to gain file access without authorization.

Introduction

As cloud computing and big data become increasingly popular, more businesses and clients are opting to outsource their files to the cloud for easy storage and management, causing cloud discs to fill up quickly. According to Cisco Global Cloud Index's "White Paper on Forecasting and Methods 2015-2020," the number of people signing up for personal cloud storage services would rise from 1.3 billion in 2015 to 2.3 billion in 2020, with global data reaching 40ZB. When confronted with such a vast amount of data, cloud service providers' ability to harness the economy and efficiency of cloud sources has become an

unavoidable obstacle. With the rise in popularity of the cloud, the problem of personal data exposure on the internet has become more prevalent. The privacy data exposure concern in the Internet of Things has steadily been uncovered as the Internet of Things have grown in popularity. There is a lot of redundant privacy data in the cloud. The risk of privacy leakage can be lowered if the redundant data can be eliminated reasonably. De-duplication technology, which uses randomly chosen strings or hash values as labels to avoid redundantly uploading identical data, is frequently used by cloud service providers to increase storage availability and reduce administrative costs. Relevant procedures can be classified into two categories based on the phase in which de-duplication is performed.

(1) Client-side de-duplication: Users just upload their data to the cloud, and the server detects and removes reduplicated data on its own. Though this proposal preserves the de-duplication capabilities, it will take a lot of bandwidth to transmit superfluous data.

(2) Client-assisted de-duplication: Before uploading, the client computes the data's hash function for cloud retrieval. The cloud server checks its local storage system for the same label after receiving the hash value. If this is the case, the server e, the server will instruct the client to terminate the upload and label them as an invalid client. The data's owner because hash values are constantly truncated, this strategy could save a lot of storage space as well as transmission time.

2. LITERATURE SURVEY

Existing Solutions

In the Existing systems once the hash code is generated with the SHA algorithm then they will encrypt the file with the same hash code, and they add some random number to the hash code then they encrypt that hash code to store on the cloud and to ensure that the hash code is not stolen by anyone. so they store that encrypted hash code on the cloud. So the problem in this system is if the hash code is known to any hacker by randomly generating some hash codes or if he knows that file then he can get the data. So there are a lot of dictionary attacks can be possible in this system.

Cloud computing allows users to access resources such as networks, apps, and services on-demand and from anywhere. As a result, a large number of businesses and individual users are outsourcing their data to cloud servers. As a result, cloud servers' data volume is rapidly increasing. Major security concern in cloud computing is how to efficiently manage the ever-increasing data. In recent years, secure de-duplication approaches have piqued the interest of both academics and industry.

• To overcome the aforementioned flaw, the DupLESS concept was proposed, in which the user generates the convergent key with the assistance of a key server. When the key server is corrupted by the cloud server, we claim that DupLESS does not work.

• In Another research, they presented a new threshold blind signature-based multi-server-aided deduplication strategy that effectively resists collusion attacks between the cloud server and several key

servers. Furthermore, we demonstrate that our structure is capable of achieving the requisite security features. De-duplication is used by cloud storage service providers such as Dropbox, Mozy, and others to save space by only storing one copy of each file uploaded. clients should encrypt their files to store them on the cloud, But previous savings will be lost.

• Message-locked encryption (of which convergent encryption is the most prominent manifestation) alleviates this tension. It is, however, vulnerable to brute-force attacks that can recover files from a known set. they used architecture for secure deduplicated storage that is resistant to brute-force attacks, which we implement in the DupLESS system.

• DupLESS uses an oblivious PRF protocol to encrypt messages using message-based keys supplied from a key-server. It allows clients to store encrypted data without losing the existing service so it stores along with the encrypted data, and it has the service perform de-duplication, and still maintain high levels of confidentiality. We show that utilizing encryption for de-duplicated storage can produce speed and space reductions that are comparable to those achieved when using the storage service with un-encrypted data.

2.2 Related Work

The user encrypts files with the created hash code in the existing system, it is not very secure because if an unauthorized person uploads the identical file value, the hash code will be quickly discovered and he will be able to view the file. To solve this problem, we've devised a blind signature technique that can't be decrypted and can't be decoded even if the individual knows the hash code. As a result, we encrypt the file with the signature rather than the hash code on our system.

3. DESIGN OF THE PROPOSED SYSTEM

In the proposed system De-duplication technology is used, which uses hash values as keys to avoid redundancy of uploading the same data or files in the cloud, and the cloud is most frequently used by cloud service providers to enhance storage availability and lower administrative costs. The client calculates the hash code for cloud retrieval before uploading. The cloud server checks its local storage system for the same hash code after receiving the hash value. If this is the case, the server will inform the client to cancel the upload and mark the client as the owner of the material. This strategy could considerably reduce both storage space and transmission overhead because hash values are always truncated. The current system is unsafe because if an unauthorized individual uploads the same file value, the hash code will be easily discovered and he will be able to access the file. To solve this problem, we've devised a blind signature technique that can't be decrypted and can't be decoded even if the individual knows the hash code. As a result, we encrypt the file with the signature rather than the hash code on our system.

System Architecture

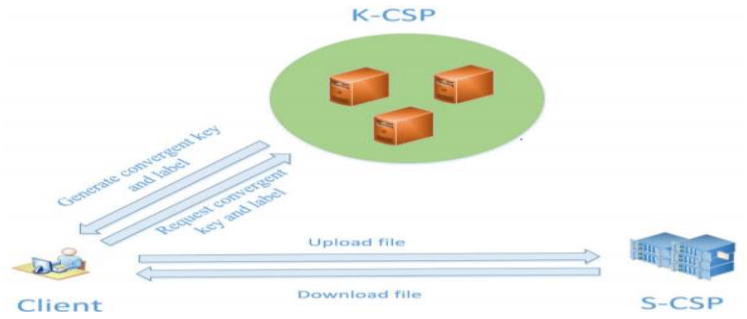

Fig 1:- System architecture representing the flow

Data Flow Diagram:

A data-flow diagram is a diagram that shows how data flows through a system or process. The DFD also contains information on the outputs and inputs of each entity, as well as the process itself. In a data-flow diagram, there are no decision rules or loops, hence there is no control flow.

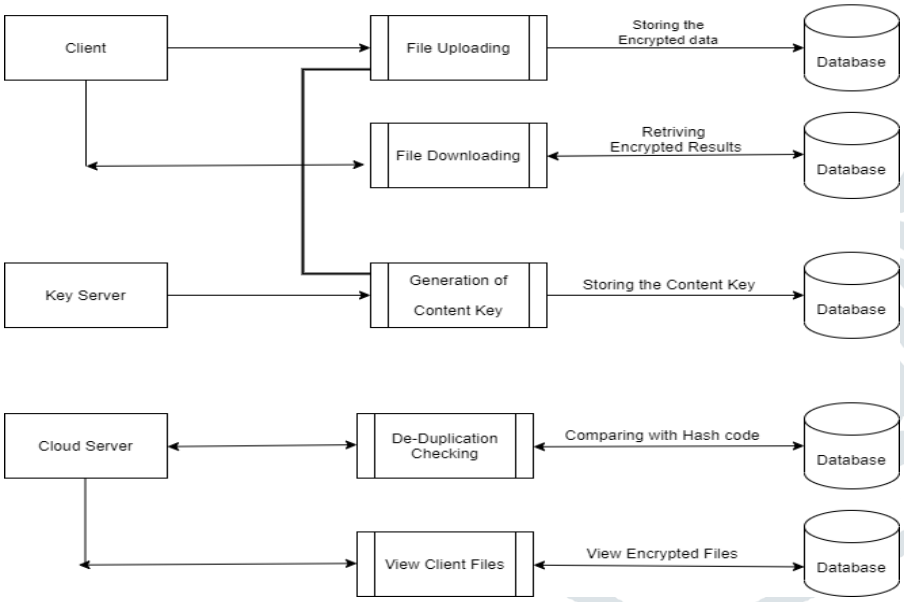

Fig 2:- Data flow diagram

Use case:

A use-case diagram depicts the behavior of a system graphically. These diagrams show a high-level view of how the system is used from the perspective of an outsider (actor).

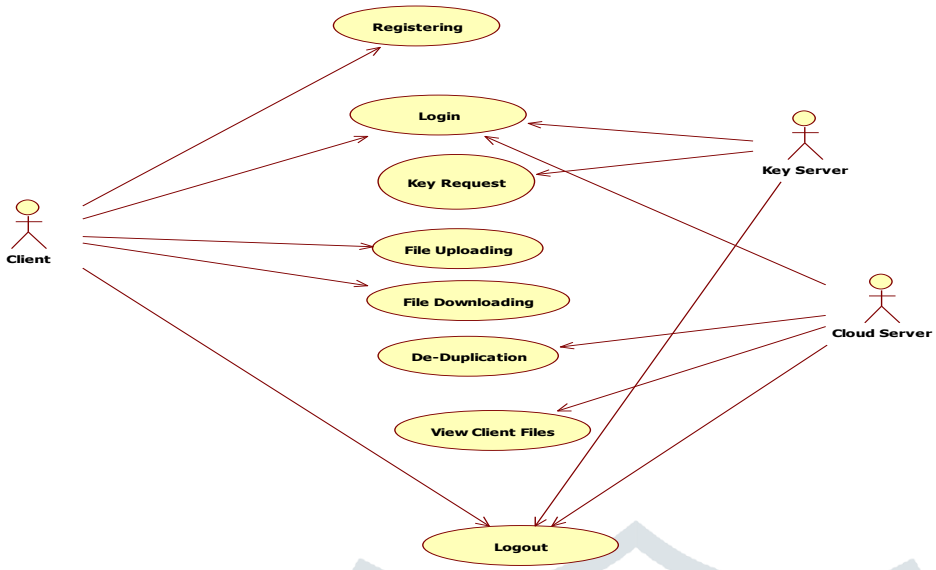

Fig 3:- Use Case

In the implementation of the proposed system, all the testing processes have been explained.

4.1 Algorithms

We used 4 Algorithms :

4.1.1.SHA Algorithm for Hash Code Generation

1. Encode the Input to binary using UTF-8 and append a single '1' to it.
2. Prepend that binary to the message block.
3. Append the original message length at the end of the message block as a 64-bit integer
4. Add 447 zeros between the encoded message and length integer so that the message block is a multiple of 512

4.1.2 Blind signature algorithm for Signature Verification

1. The user divides the file into n blocks according to a function of fixed permutation. Then it calculates H in terms of a hash function that we defined before (user defines a hash function before uploading them to the cloud) and they deliver information to their corresponding key servers. First, the user generates a hash code.
2. Once After receiving H, the key server figures out according to its private but fixed random string and broadcasts ri to the user and also to a subgroup containing at least t servers. The servers who will not participate in the following procedure should also send their ui secretly to the client. Then that hash code is sent to the key server.
3. The client randomly samples ϕ from Z q and broadcasts h' to all servers.
4. After all, the before-mentioned information is collected, and each user who is part of the cloud signs h' with their private key. The key server adds a random number and generates the signature.

5. Finally, returns the signature to the client.

4.1.3.Bloom Filter Algorithm used for Proof of Verification

1. Create an Empty Bloom filter of a bit array of m bits all set to Zero
2. Take the k1 number of Hash Functions to calculate the hash codes for a given input data or file. When we want to add an item to the filter, the bits at k1 indices are calculated using hash functions.
3. Set the bits for indices generated using hash functions to 1
4. If you enter anything again then calculate hashes similarly
5. Now check whether a file is present in the filter or not

- Calculate all the hashes using h1,h2,h3 respectively
- Check all these indices are set to '1' or not in the bit array
  - If all indices are set to '1' then the file is present
  - Else if any of the bits are set to '0' then definitely file is not present

4.1.4 AES Algorithm

Encryption Process:

1. Byte Substitution (SubBytes)
2. Shift rows
3. MixColumns
4. Addroundkey

4.2 Testing Process

What is Waterfall Model?

The Waterfall Model is a step-by-step process that separates software development into phases. During the SDLC phase, each step is meant to execute specific tasks. Winston Royce first debuted it in 1970.

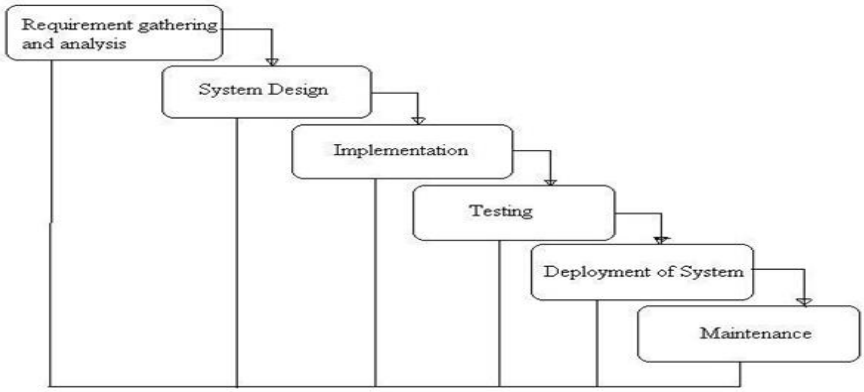

Fig 4:- Explaining the process of a waterfall model

Browser Compatibility Testing to Application

Browser Compatibility is how a web page looks in different web browsers. Different browsers read the website code differently. In other words, Chrome will render a website Differently than Firefox or Internet Explorer will. Cross-browser compatibility testing has been gaining a lot of traction in recent years and there is a reason for it. While technology is evolving rapidly, people aren't. A significant amount of people are resistant to changes, or more specifically, "have an aversion to upgrading their tech". In this scenario, it's browser compatibility testing that enables companies to ensure that no customer is left behind or has an experience that is not desired. So even though browsers like Google Chrome and Firefox dominate the market, people are using their older versions, or other browsers. And their numbers are too high to be ignored.

What is cross-browser compatibility testing?

Cross-browser compatibility testing is a non-functional form of testing, which emphasizes availing your website's basic features and functionality to users on different browser-OS combinations, devices, and assistive tools.

How does it impact your application?

Not all browsers and devices work on the same configuration; they face browser compatibility issues on different levels. This inconsistency is the reason why you might observe the lack of application uniformity across browsers and devices. You would not want a section of your prospective users to not be able to access the application features. That is what makes cross-browser testing important. If your website is not tested and debugged on different platforms and browsers, it won't work the same on all of them, causing inconvenience to the users, and subsequently impacting your business.

Which browsers to choose for cross-browser testing?

Since it's impossible to test on every possible browser-device combination, you need to shortlist the most important ones to test your web application on. As of December 2018, Google Chrome has the largest number of users. It accounts for about 70.95% of the market. Firefox comes second with a market share of 10.05%, while others such as IE, Safari, and Edge have a market share of approx 4-5% each.

Fig 5:- Result of the Application Home page

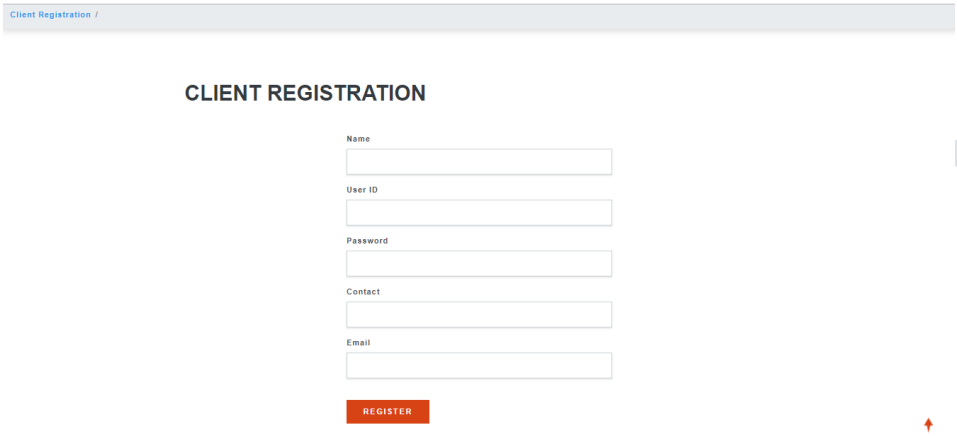

Fig 6:- Client registration

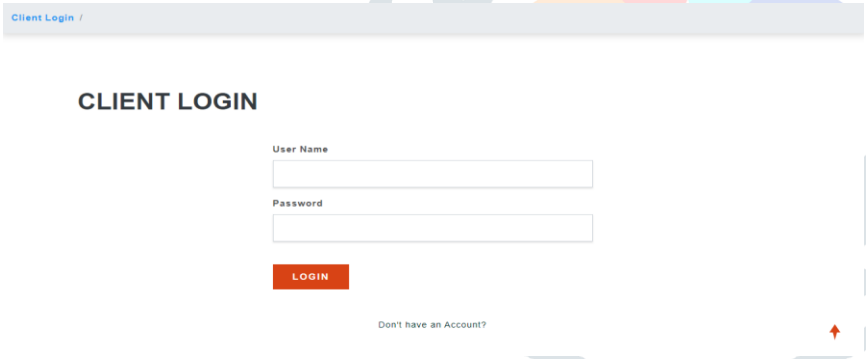

Fig 7:- Client login

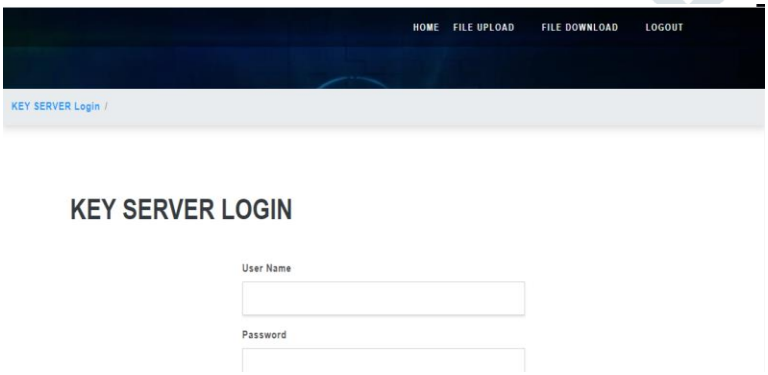

Fig 8:- Key Server Login

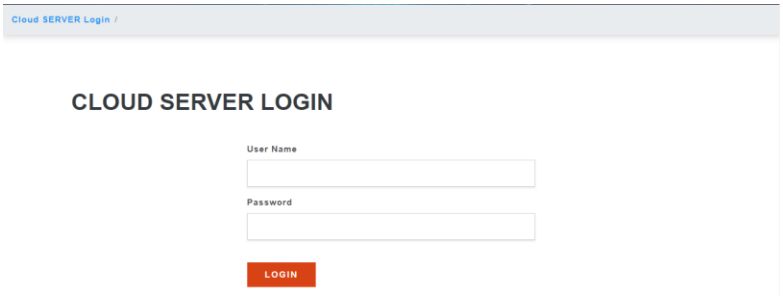

Fig 9:- Cloud Server login

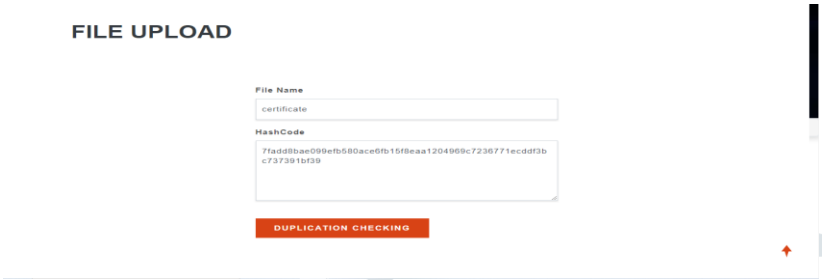

Fig 10:- File Upload

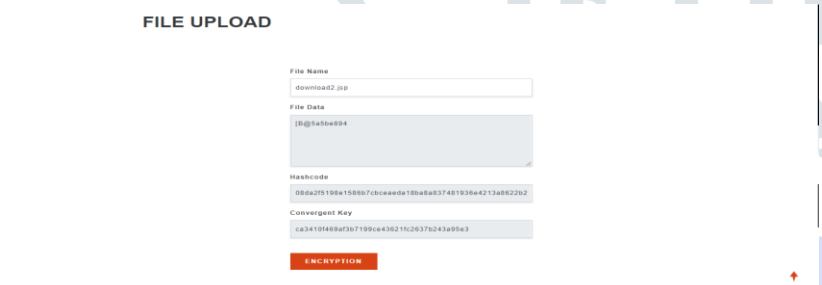

Fig 11:-Convergent Key to encrypt the file in the File uploading page

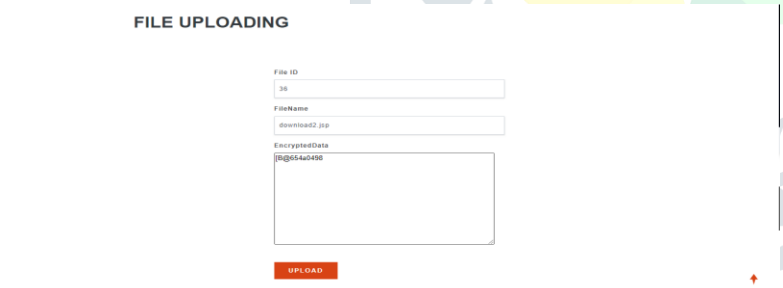

Fig 12:-File Uploading page after the encryption page

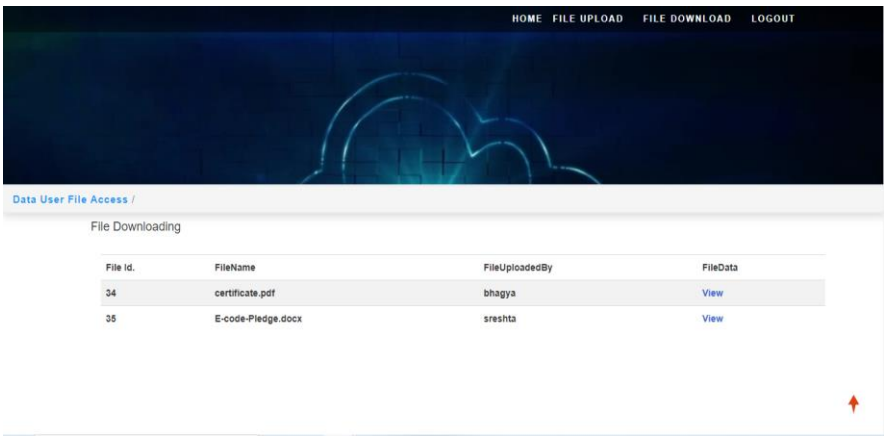

Fig 13:- File Downloading

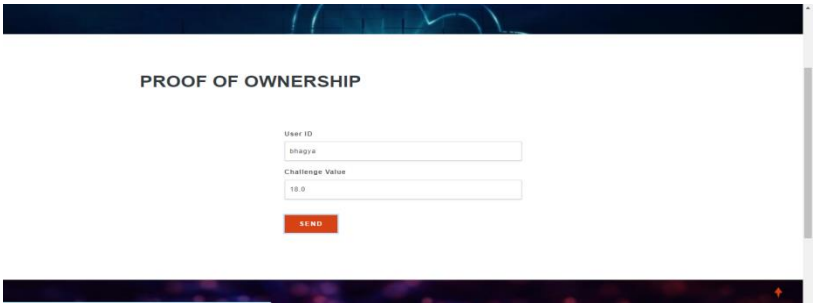

Fig 14:-Sending Request to a cloud server for proof of verification

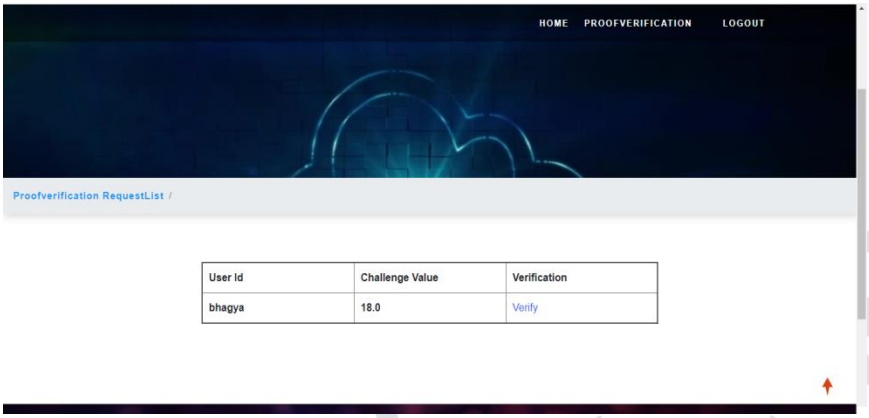

Fig 15:- View User Request in the cloud

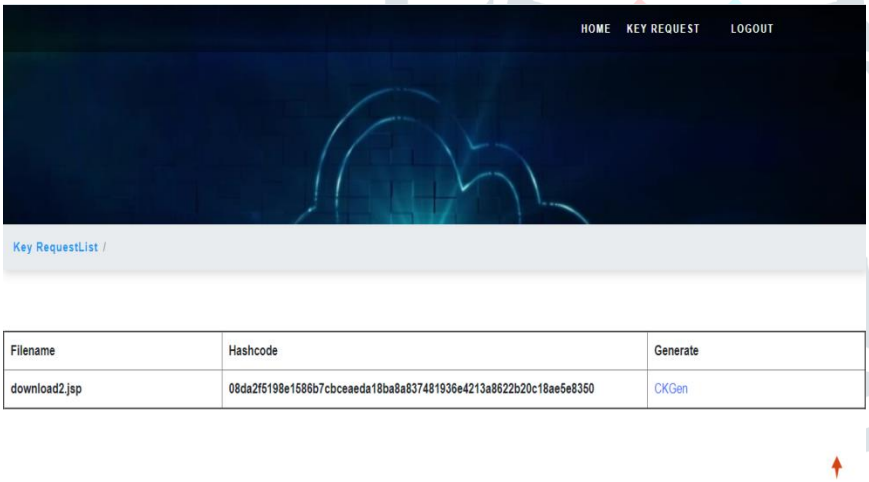

Fig 16:- Key Request to Key server

Conclusion

Cloud storage service providers cater to the needs of enterprises and people by allowing them to store, transmit, and backup their ever-increasing amounts of data at a low cost while also offering access to other cloud resources. Cloud service providers use the most extensively used deduplication approach to provide effective data storage because it allows for the storage of a single instance of data while limiting the storage of duplicate copies, reducing storage overhead and upload bandwidth. Customers who upload data to the cloud are particularly concerned about data security and confidentiality. Therefore, clients have encrypted the files and outsourced them to a cloud server. Previous research (*Vidhya*, 2017*)* proposed Convergent Encryption to encrypt the cloud files based on the generation of hash code from the file content. So, while file de-duplication, if the malicious users know the hash code as the content key, then

cloud files will be compromised by attackers, therefore in this proposed system we are implementing the Blind Signature methodology, which will generate the signature as hash code from the content key and used for file encryption then even the attacker known the content key, they unable to compromise the cloud files. As well, we had implemented the Bloom filter technique where the client gets de-duplication then they need to verify with the cloud server by generating the proof value, if the proof is valid then they can get the reference of matching cloud file. The concept of Attribute-Based Encryption and Identity Based Encryption can be combined to provide secure Data De-duplication to provide better security with less overhead.