IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CHARON: A NOVEL PROTOCOL FOR SECURE DATA STORAGE AND SHARING IN CLOUD -OF- CLOUD ENVIRONMENT

NAMGEDDA SIREESHA [#1], L. SOWJANYA [#2]

[#1] MCA Student, Master of Computer Applications,

D.N.R. College, P.G.Courses & Research Center, Bhimavaram, AP, India.

[#2] Assistant Professor, Master of Computer Applications,

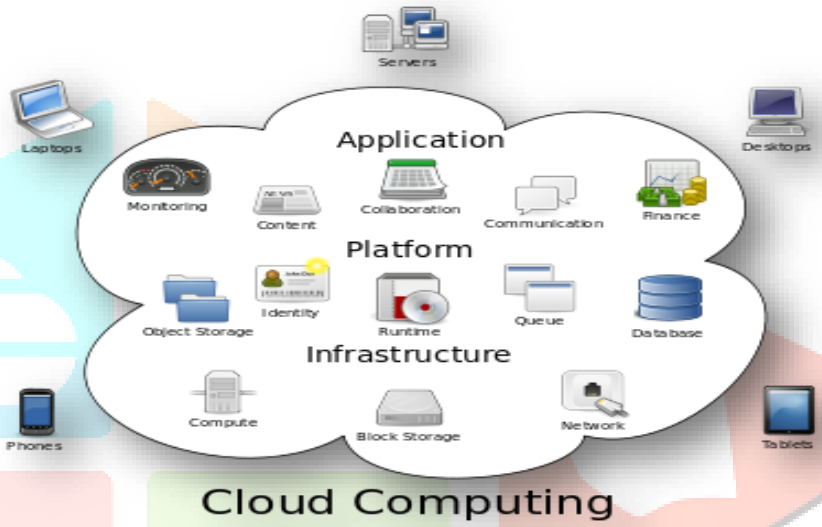
D.N.R. College, P.G.Courses & Research Center, Bhimavaram, AP, India.

Abstract

Now a day's almost all small scale and large scale organizations try to adopt the centralized cloud server for their data storage and accessing from the remote locations connected all together from a centralized server with the help of internet. As we all know that till now no cloud service provider is providing privacy for the data in terms of encryption and key in order to provide access data authorization. In this paper, we present CHARON, a cloud-backed storage system capable of storing and sharing big data in a secure, manner for sharing the sensitive data. Here we try to divide the cloud sever into two factor access control in which one control is used for storing the sensitive documents in an encrypted manner and another control is to share the key permissions which is requested by the data users. By conducting various experiments on our proposed model, our result clearly tells that our proposed system is practical and efficient for storing data in a secure manner.

1. INTRODUCTION

Cloud Computing is the utilization of processing assets (equipment and programming) that are conveyed as an administration over a system (normally the Internet). The name originates from the regular utilization of a cloud-formed image as a deliberation for the perplexing foundation it contains in framework outlines. Distributed computing endows remote administrations with a client's information, programming and calculation. Distributed computing comprises of equipment and programming assets made accessible on the Internet as oversaw outsider administrations. These administrations regularly give access to cutting edge programming applications and top of the line systems of server PCs.



Structure of cloud computing

How Cloud Computing Works?

The objective of distributed computing is to apply customary supercomputing, or superior registering power, ordinarily utilized by military and research offices, to perform several trillions of calculations for every second, in buyer situated applications, for example, money related portfolios, to convey customized data, to give information stockpiling or to influence vast, vivid PC amusements.

The distributed computing utilizes systems of extensive gatherings of servers ordinarily running ease purchaser PC innovation with specific associations with spread information preparing errands crosswise over them. This mutual IT framework contains huge pools of frameworks that are connected together. Regularly, virtualization methods are utilized to augment the intensity of distributed computing.

Characteristics and Services Models:

The remarkable qualities of distributed computing dependent on the definitions given by the National Institute of Standards and Terminology (NIST) are laid out beneath:

- **On-request self-administration**: A purchaser can singularly arrangement processing capacities, for example, server time and system stockpiling, as required naturally without requiring human association with each specialist co-op's.
- **Broad organize get to**: Capabilities are accessible over the system and got to through standard instruments that advance use by heterogeneous slim or thick customer stages (e.g., cell phones, PCs, and PDAs).
- **Resource pooling**: The supplier's registering assets are pooled to serve numerous purchasers utilizing a multi-occupant display, with various physical and virtual assets powerfully doled out and reassigned by shopper request. There is a feeling of area autonomy in that the client by and large has no control or information over the definite area of the gave assets yet might most likely determine area at a larger amount of deliberation (e.g., nation, state, or server farm). Instances of assets incorporate capacity, preparing, memory, organize transmission capacity, and virtual machines.
- **Rapid versatility**: Capabilities can be quickly and flexibly provisioned, now and again consequently, to rapidly scale out and quickly discharged to rapidly scale in. To the shopper, the abilities accessible for provisioning regularly seem, by all accounts, to be boundless and can be obtained in any amount whenever.
- **Measured administration:** Cloud frameworks consequently control and improve asset use by utilizing a metering capacity at some dimension of deliberation fitting to the kind of administration (e.g., capacity, handling, data transfer capacity, and dynamic client accounts). Asset utilization can be overseen, controlled, and revealed giving straightforwardness to both the supplier and buyer of the used administration.

SIGNIFICANCE OF THE WORK

As we all know that till now no cloud service provider is providing privacy for the data in terms of encryption and key access in order to provide data authorization. We present CHARON, a cloud-backed storage system capable of storing and sharing big data in a secure, reliable, and efficient way using multiple cloud providers and storage repositories to comply with the legal requirements of sensitive personal data.

2. LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, ten next steps are to determine which operating system and language used for developing the tool. Once the programmers start building the tool, the programmers need lot of external support. This support obtained from senior programmers, from book or from websites. Before building the system the above consideration r taken into for developing the proposed system.

1) BYZANTINE DISK PAXOS: OPTIMAL RESILIENCE WITH BYZANTINE SHARED MEMORY

We present Byzantine Disk Paxos, an asynchronous shared-memory consensus algorithm that uses a collection of n > 3t disks, t of which may fail by becoming non responsive or arbitrarily corrupted. We give two constructions of this algorithm; that is, we construct two different t-tolerant (i.e., tolerating up to t disk failures) building blocks, each of which can be used, along with a leader oracle, to solve consensus. One building block is a t-tolerant wait-free shared safe register. The second building block is a t-tolerant regular register that satisfies a weaker termination (liveness) condition than wait freedom: its write operations are wait-free, whereas its read operations are guaranteed to return only in executions with a finite number of writes. We call this termination condition finite writes (FW), and show that wait-free consensus is solvable with FW terminating registers and a leader oracle. We construct each of these ttolerant registers from n > 3t base registers, t of which can be non-responsive or Byzantine. All the previous t-tolerant wait-free constructions in this model used at least 4t + 1 fault-prone registers, and we are not familiar with any prior FW terminating constructions in this model.

2) UNIDRIVE: SYNERGIZE MULTIPLE CONSUMER CLOUD STORAGE SERVICES

Consumer cloud storage (CCS) services have become popular among users for storing and synchronizing files via apps installed on their devices. A single CCS, however, has intrinsic limitations on networking performance, service reliability, and data security. To overcome these limitations, we present UniDrive, a CCS app that synergizes multiple CCSs (multi-cloud) by using only few simple public RESTful Web APIs. UniDrive follows a server-less, client-centric design, in which synchronization logic is purely implemented at client devices and all communication is conveyed through file upload and download operations. Strong consistency of the metadata is guaranteed via a quorum-based distributed mutual-exclusive lock mechanism. UniDrive improves reliability and security by judiciously distributing erasure coded files across multiple CCSs. To boost networking performance, UniDrive leverages all available clouds to maximize parallel transfer opportunities, but the key insight behind is the concept of data block over-provisioning and dynamic scheduling. This suite of techniques masks the diversified and varying network conditions of the underlying clouds, and exploits more the faster clouds via a simple yet effective in-channel probing scheme. Extensive experimental results on the global Amazon EC2 platform and a real-world trial by 272 users confirmed significantly superior and consistent sync performance of UniDrive over any single CCS .

3) PERM: Practical reputation-based blacklisting without TTPS

AUTHORS: M. H. Au and A. Kapadia

A few clients may get out of hand under the front of secrecy by, e.g., ruining pages on Wikipedia or posting disgusting remarks on YouTube. To counteract such maltreatment, a couple of unknown qualification plans have been suggested that renounce access for getting out of hand clients while keeping up their secrecy to such an extent that no confided in outsider (TTP) is associated with the disavowal procedure. As of late we proposed BLACR, a without TTP plot that bolsters 'notoriety based boycotting' - the specialist organization can score clients' mysterious sessions (e.g., great versus wrong remarks) and clients with deficient notoriety are denied get to. The significant disadvantage of BLACR is the direct computational overhead in the extent of the notoriety list, which enables it to help notoriety for just a couple of thousand client sessions in down to earth settings. We propose PERM, a disavowal window-based plan (mischievous activities must be gotten inside a window of time), which makes calculation free of the measure of the notoriety list. PERM in this manner bolsters a huge number of client sessions and makes notoriety based boycotting reasonable for vast scale organizations.

4) BLACR: TTP-free blacklistable anonymous credentials with reputation

AUTHORS: M. H. Au, A. Kapadia, and W. Susilo

Unknown confirmation can give clients the permit to get into mischief since there is no dread of retaliation. As a hindrance, or intends to denial, different plans for responsible namelessness include some sort of (conceivably disseminated) confided in outsider (TTP) with the ability to recognize or interface acting mischievously clients. As of late, plans, for example, BLACK and PEREA indicated how unknown denial can be accomplished without such TTPs—mysterious clients can be renounced in the event that they act up, but then no one can recognize or connection such clients cryptographically. Regardless of being the cutting edge in unknown denial, these plans permit just a fundamental type of renouncement adding up to 'deny anyone with d or more mischievous activities' or 'disavow anyone whose joined mischief score is excessively high' (where mischievous activities are alloted a 'seriousness' score). We present BLACR, which fundamentally progresses mysterious denial in three different ways:

1) It establishes a first endeavor to sum up notoriety based unknown disavowal, where negative or positive scores can be doled out to unknown sessions over various classifications. Servers can square clients dependent on approaches, which indicate a boolean mix of notorieties in these classes.

2) We present a weighted augmentation, which permits the all out seriousness score to increase for different mischievous activities by a similar client.

3) We make a huge improvement in confirmation times through a system we call express path verification, which makes notoriety based mysterious disavowal functional.

3. EXISTING SYSTEM

In the existing cloud servers ,there was no concept like encryption of cloud data and also there was no facility like key generation and maintenance of data. The current cloud storage is almost centralized and all the data which is stored along with details of data owners and data users is clearly visible by the cloud server department, which is almost a big problem in the current cloud service providers. In the current cloud servers all the data can be viewed and accessed by any one who is having an account access within the cloud, so that the data is not having integrity or security in terms of any modification or changes done by any user. Also in the current cloud servers there is no facility like multi cloud access. Also there is no concept like providing authorization for the owners and users for accessing the file. And there is also no concept to identify the attacker details.

LIMITATION OF EXISTING SYSTEM

In the existing or current clouds the following are the main limitations that are available

1. All the existing schemes are limited to the single-owner model.
2. All the current cloud servers has search in a normal manner under plain text model, but they don't have any facility to search in a ENRYPTED manner
3. The existing cloud servers are almost operated in a centralized manner, where all the access can be viewed and monitored by the cloud service providers.
4. The current cloud servers don't have a facility to store the sensitive information under a de-centralized manner in which the access should lie in the hands of separate individual departments.

4. PROPOSED SYSTEM

As we all know that till now no cloud service provider is providing privacy for the data in terms of encryption and key access in order to provide data authorization. We present CHARON, a cloud-backed storage system capable of storing and sharing big data in a secure, reliable, and efficient way using multiple cloud providers and storage repositories to comply with the legal requirements of sensitive personal data.

ADVANTAGES OF THE PROPOSED SYSTEM

The following are the advantages of the proposed system:

1. Our protocol a great flexibility for the system to set different access policies according to different scenarios.
2. At the same time, the privacy of the user is also preserved.
3. To show the practicality of our system, we simulate the prototype of the protocol.

4. Here we try to create a area for files storage in terms of cache, in which if any attacker try to create any attack on the data, the data will not be lost and it will stored safely inside the cache storage.

There is a utmost security for the data from the attackers

5. SOFTWARE PROJECT MODULES

Implementation is the stage where the theoretical design is converted into programmatically manner. In this stage we will divide the application into a number of modules and then coded for deployment. We have implemented the proposed concept on Java programming language with JEE as the chosen language in order to show the performance this proposed protocol. There are mainly 4 modules present in the application. They are as follows:

1. Data User Module
2. Authority Module
3. Trustee Module
4. Cloud server

Now let us discuss about each and every module in detail as follows:

5.1 DATA USER MODULE

- Every user need to register while accessing to cloud.
- After user registered, at the time of user login then user need to provide one time key to access user home.
- One time key will be provided by Charon 1. The key will be received to the corresponding user mail id.
- After user access the user home, User can view the all files upload in cloud.
- User need to send the file request for both trustee and authority.
- After user have the two factor access control, user can download the corresponding file.

5.2 AUTHORITY MODULE

- Authority will upload the file in cloud. And uploaded file will store in drive HQ in encrypted format.
- Authority will give secret key for all files when user request for any file and the secret key will be send to corresponding user mail Id.
- He is the one who mainly provide security of the data which is stored into the live cloud server.

5.3 TRUSTEE MODULE

- It acts as admin for cloud server.
- Trustee will give request for all files security response when user request for any file.
- He is one who is responsible for Granting Permissions for the Users during Login
- He is mainly responsible for user authentication.

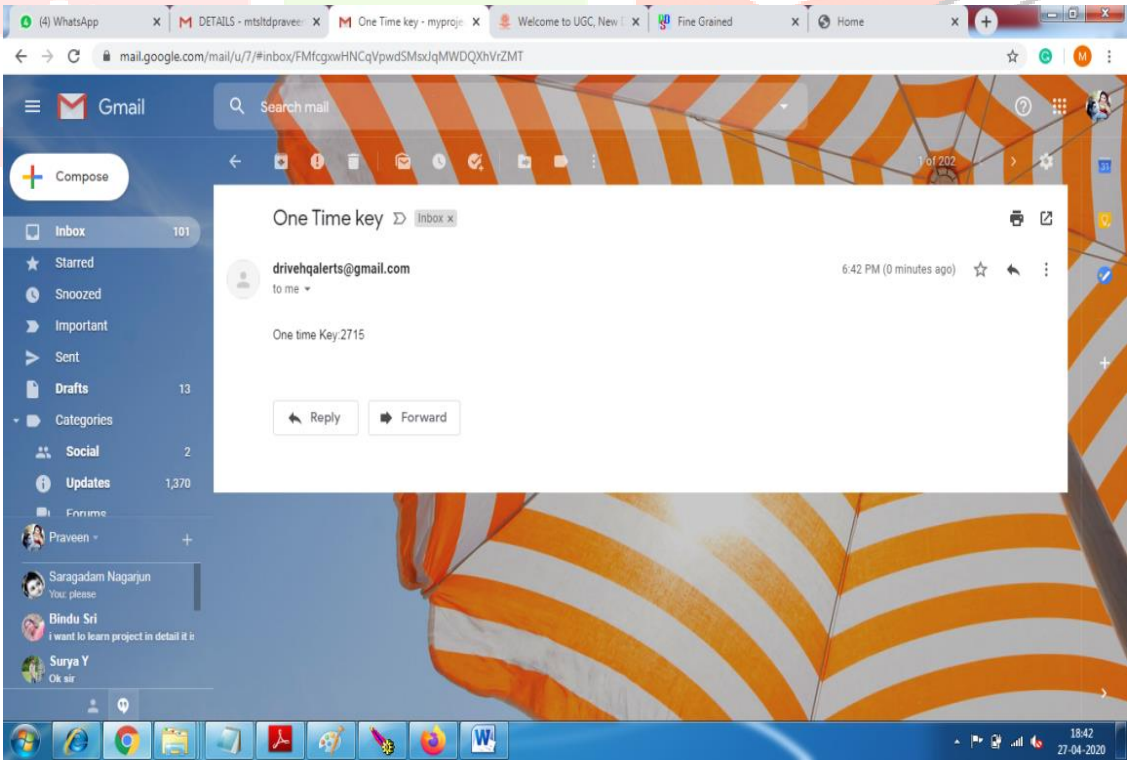
5.4 CLOUD SERVER MODULE

- Cloud view uploaded files in cloud.
- Cloud view Downloaded files by user in cloud.
- This is one which can accept all the data from authority and store the data in encrypted manner.

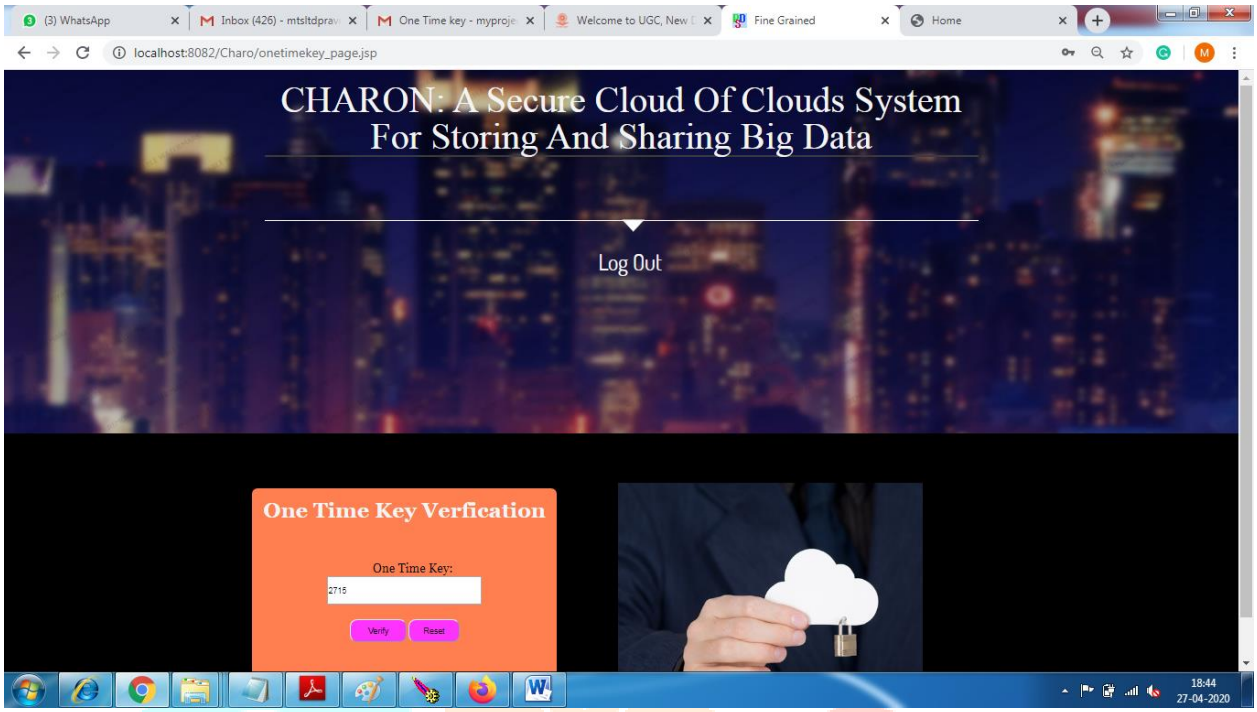
In this application we try to use this cloud as live cloud server and try to store the data under DRIVEHQ public cloud server.

6. OUTPUT SCREENS

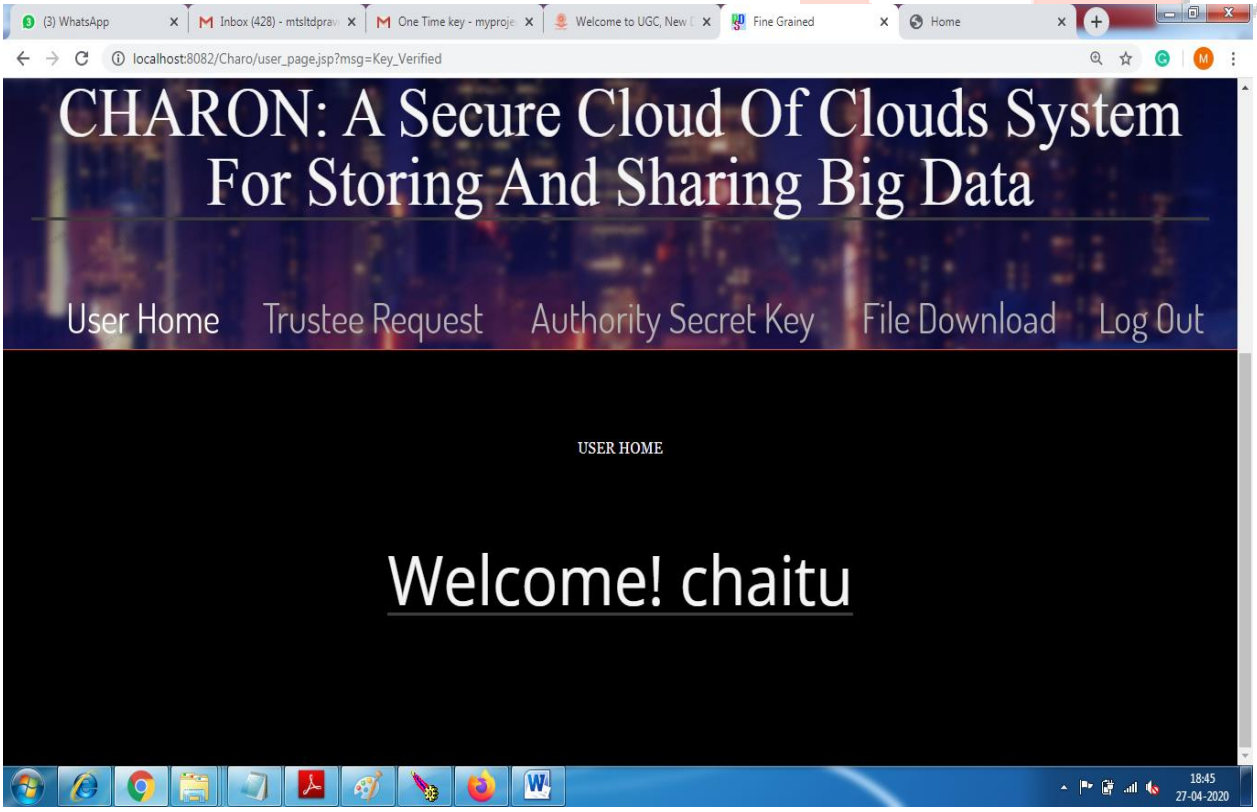
USER WILL READ OTP TO LOGIN INTO HIS ACCOUNT



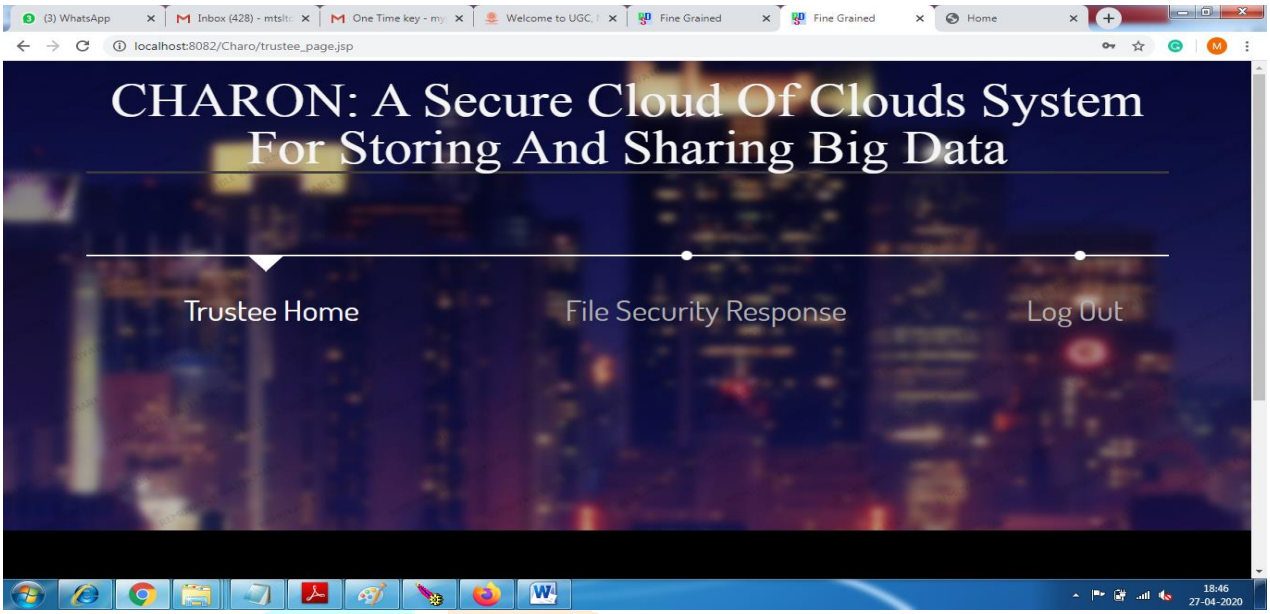
USER SUBSTITUTE OTP TO ENTER INTO HIS ACCOUNT



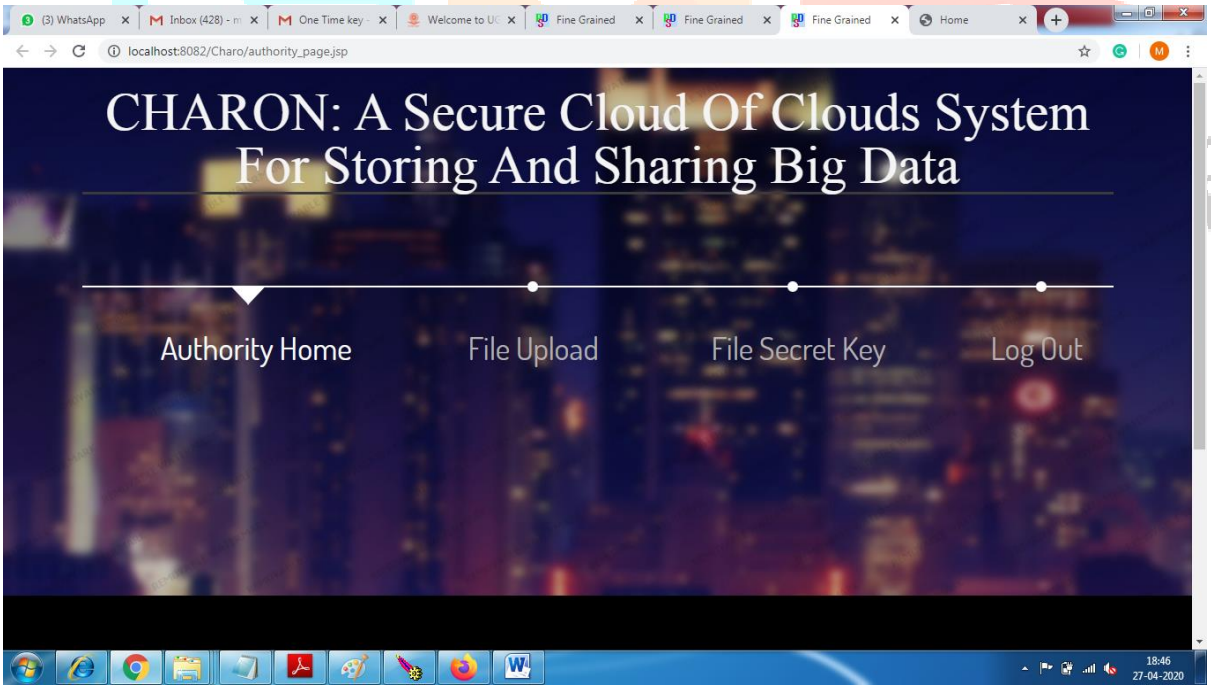
USER MAIN PAGE



TRUSTEE LOGIN FOR TO ACCESS IT

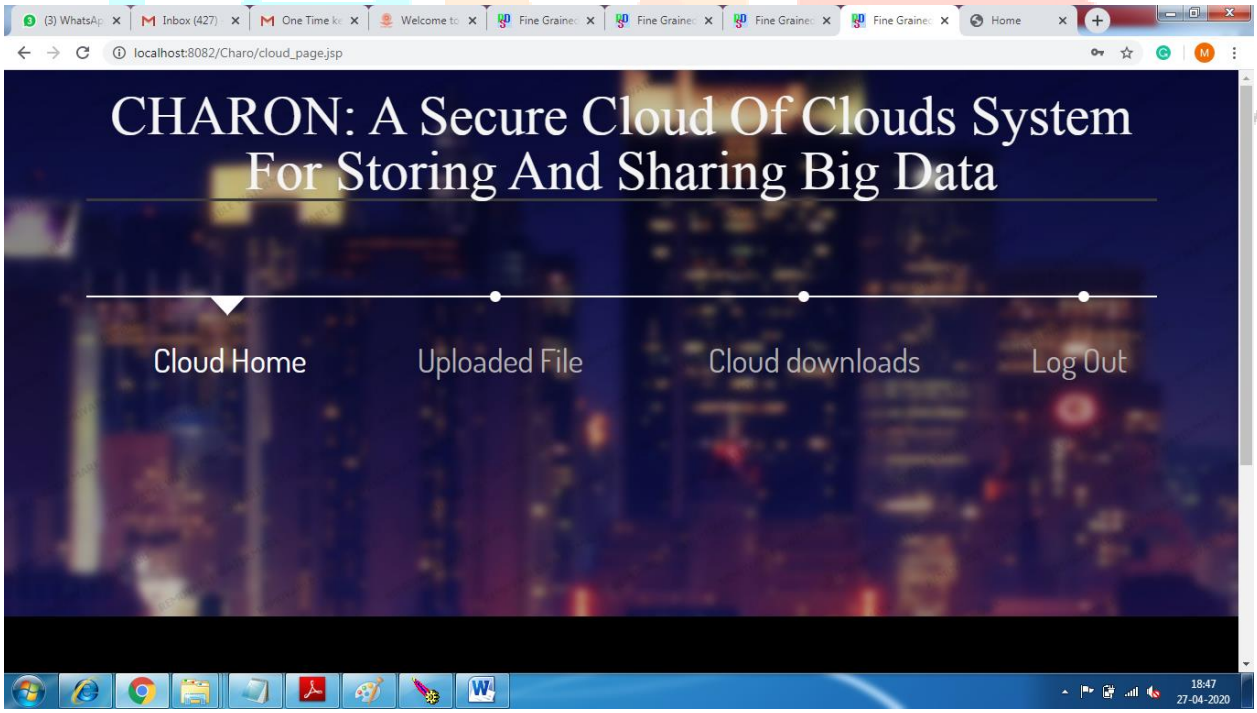


AUTHORITY LOGIN FOR TO ACCESS IT

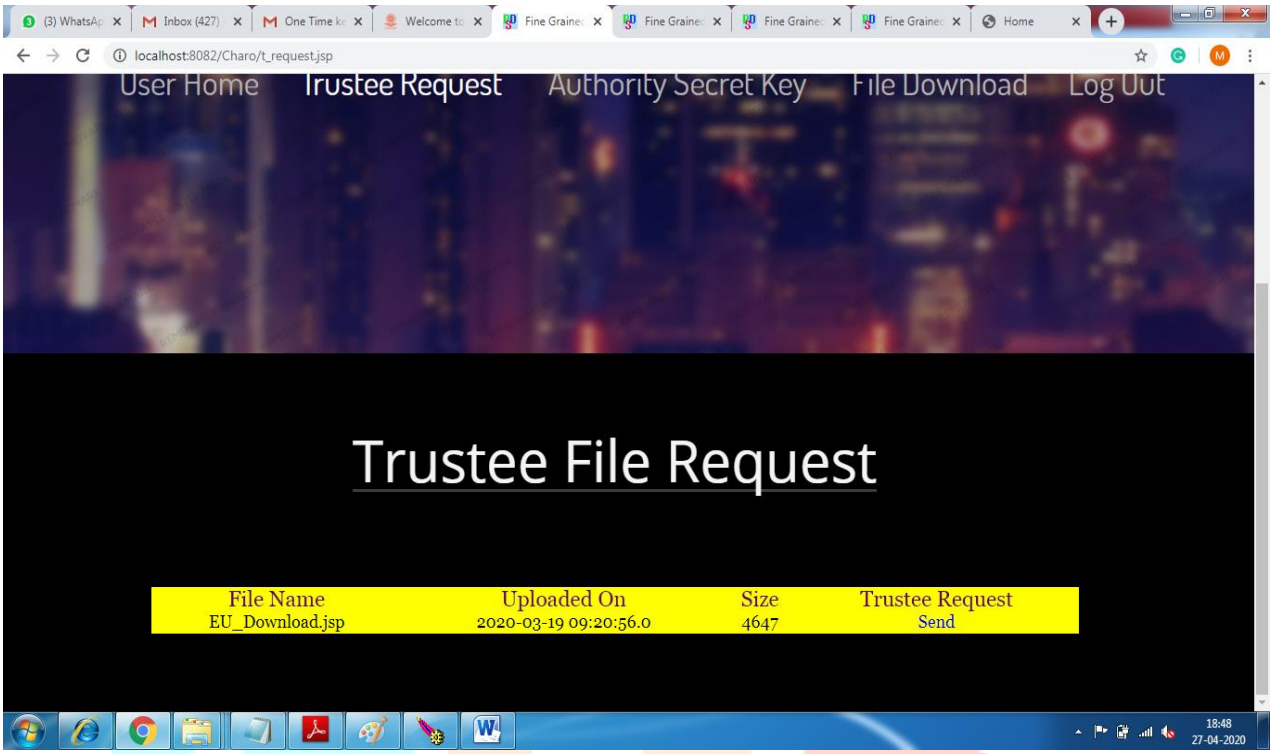


CLOUD LOGIN : SHOWS WHAT ARE THE OPERATIONS DONE ON IT

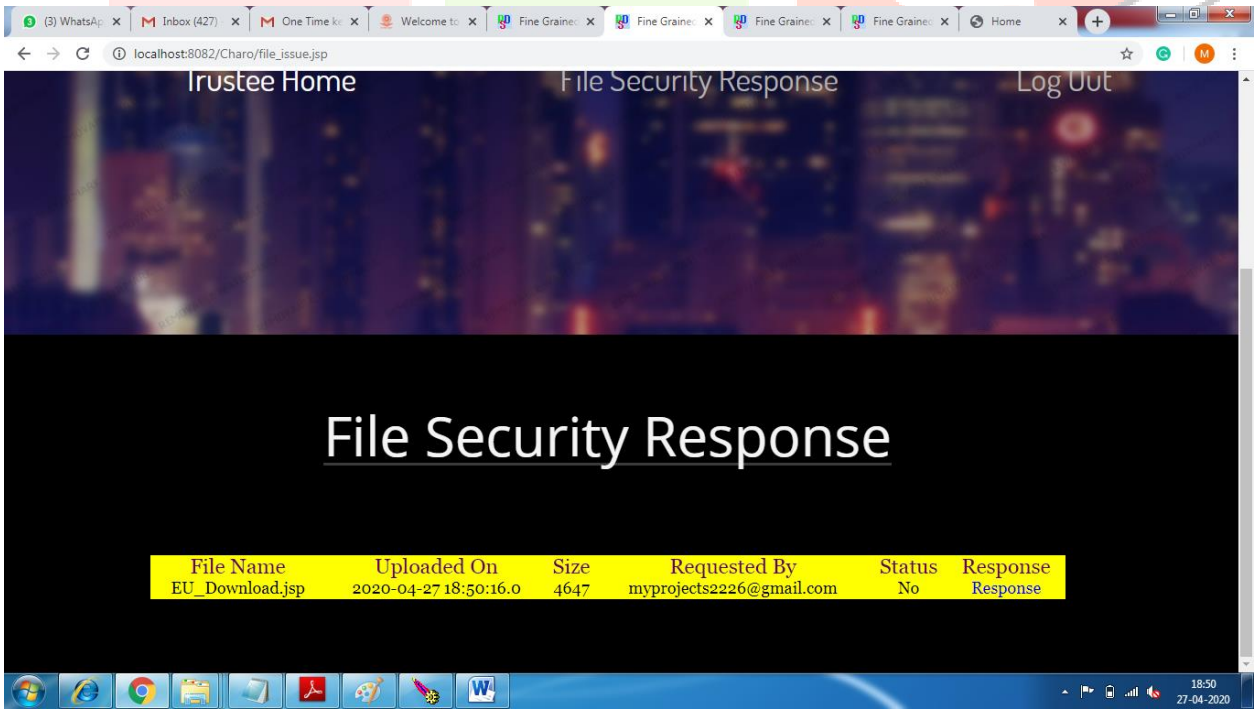
CLOUD MAIN PAGE



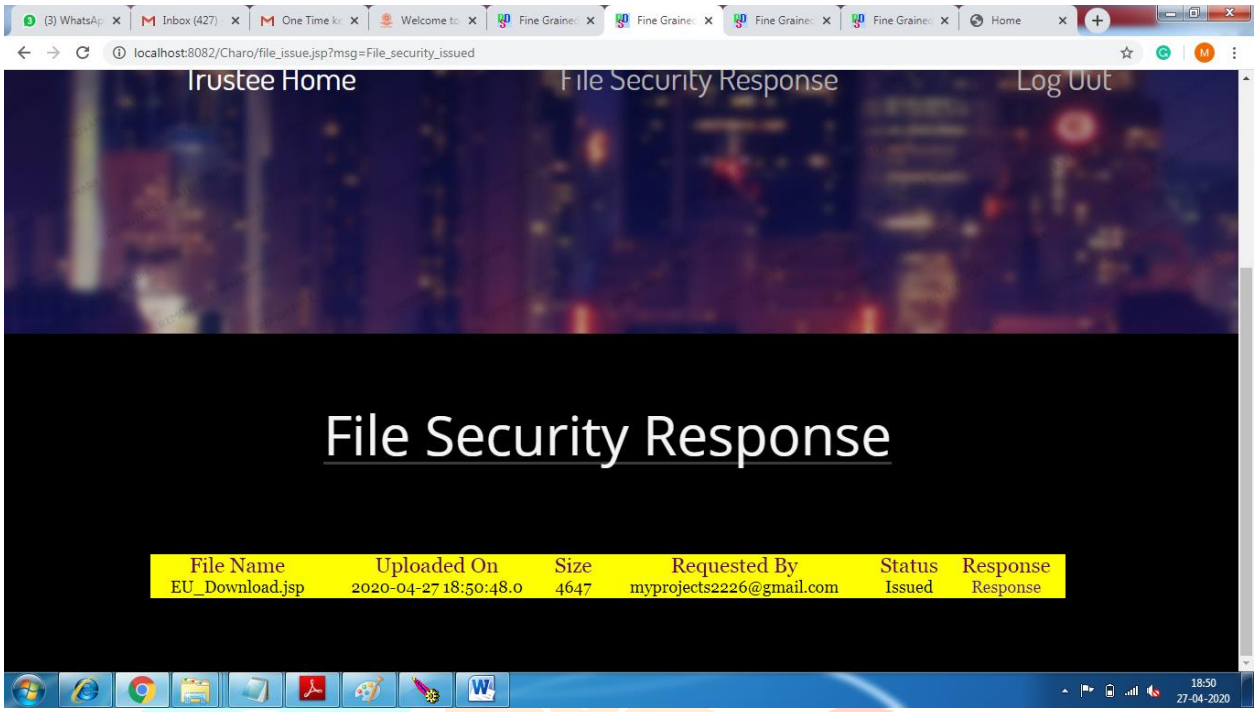
USER REQUEST TRUSTEE FOR FILE ACCESS PERMISSIONS



TRUSTEE (CHARON 1) WILL TRY TO ACCEPT THE USER REQUEST



TRUSTEE GRANTS THE PERMISSION.SO STATUS CHANGES FROM NO TO YES.



7. CONCLUSION

In this project, we have developed a new framework called as CHARON: A secure cloud of clouds system for storing and sharing big data by exhibiting the importance of cloud security. In view of the quality based access control component, the proposed CHARON access control framework has been recognized to not just empower the cloud server to confine the entrance to those clients with a similar arrangement of traits yet in addition protect client security. Through execution assessment, we showed that the development is "attainable". We leave as future work to additionally improve the proficiency while keeping every single pleasant element of the framework

8. REFERENCES

1) R. A. Popa, F. H. Li, and N. Zeldovich, "An ideal-security protocol for order- on Security and Privacy (SP'13). IEEE, 2013, pp. 463–477.

2) J.-M. Bohli, N. Gruschka, M. Jensen, L. L. Iacono, and N. Marnau, "Security and privacy-enhancing multicloud architectures," IEEE Transactions on Dependable and Secure Computing, vol. 10, no. 4, pp. 212– 224, 2013.

3) Y. Elmehdwi, B. K. Samanthula, and W. Jiang, "Secure k-nearest neighbor query over encrypted data in outsourced environments," in 2014 IEEE 30th International Conference on Data Engineering. IEEE, 2014, pp. 664–675.

4) F. Hao, J. Daugman, and P. Zielinski, "A fast search algorithm for a large fuzzy database," IEEE Transactions on Information Forensics and Security, vol. 3, no. 2, pp. 203–212, 2008.

5) A. Castelltort and A. Laurent, "Fuzzy queries over NoSQL graph databases: perspectives for extending the cypher language," in International Conference on Information Processing and Management of Uncertainty in Knowledge-Based Systems. Springer, 2014, pp. 384– 395.

6) J. Li, Q. Wang, C. Wang, N. Cao, K. Ren, and W. Lou, "Fuzzy keyword search over encrypted data in cloud computing," in Proceedings of the 29th IEEE International Conference on Computer Communications(INFOCOM2010). IEEE, 2010, pp. 1–5.