Volume 7, Issue 3, March 2017 ISSN: 2277 128X
International Journal of Advanced Research in Computer Science and Software Engineering
Research Paper
Available online at: www.ijarcsse.com

Data Security System in Cloud by Using Fog Computing and Data Mining

Parashar Sangle, Romit Deshmukh, Rohit Ghodake, Akash Yadav, Prof. Jitendra Musale
Department of Computer Engineering, Abmsp's Anantrao pawar College of Engineering & Research, Pune, Maharashtra, India

DOI: 10.23956/ijarcsse/V7I3/0119

Abstract: ***Cloud computing can be defined as a group of computers and servers which are connected together on a network. Now-a-days, as many organizations and enterprises are beginning to adopt IOT (internet of things), they all need large amount of data to be accessed quickly, to ensure safety on cloud from attacker or insider, it is important to secure cloud data; this can prove helpful in government organization or any IT-industry. In any case if the existing encryption data protection mechanism fails in preventing data theft attacks from criminal attacker, especially from insider to the cloud provider; to provide security to cloud data from unauthorized access from malicious attacker we propose a different approach to secure the data in cloud system by using decoy technology. We monitor user behavior or data access patterns in cloud system and identify abnormal data access patterns. When any unauthorized data access pattern is suspected then the decoy data is provided to the unauthorized user. This mechanism ensures protection against the misuse of the user's real data. In any case if the real user gets trapped in this system then user can challenge the system asking for one-time password (OTP) for verification.***

Keywords: Fog computing, Data mining, Clouding computing, Cloud security.

I. INTRODUCTION

Cloud computing is a group of computers and servers that are connected together over the internet. Today many small-scale or large-scale organizations as well as many enterprises use cloud to store large amount of data, the data might be private data or business information. There is a need of large amount of data to be accessed faster and locally and is ever growing. This is where the fog computing comes into picture.

Fog computing is a term created by Cisco. Fog computing, also known as fog networking, it is a distributed infrastructure in which certain application services are managed at the edge of the network by using device and other still managed in the cloud. Basically it is a middle layer between the cloud and hardware or user end devices, which provides efficient analysis, data processing and storage. The goal of fog computing is to improve efficiency and reduce the amount of data that needs to be transported to the cloud for data processing, analysis and storage.

If attackers are intelligent and launch attack against cloud system, then it is easy to break cloud user's password or if attacker is an insider then it is possible to steal someone's user ID and password easily and try to get unauthorized access of cloud system in order to steal private or business information of a particular user. To overcome this problem, we propose different technique to provide security to cloud data from unauthorized user by creating confusion by using decoy technology. That we have come to call fog computing. We can use this technology to launch disinformation attacks against unauthorized user or insider thereby preventing them the access to real user data. In this paper the system monitors user behavior activity or real user data access patterns, if any abnormal data access patterns are suspected then fog computing launches this disinformation attack against unauthorized user. In this decoy data base are filled with fake information when any abnormal data access patterns are identified by system then fake data from decoy database is provided to the invalid user. By using this technology, we can secure original cloud data from attackers and also protects misuse of real user information.

II. PROPOSED SYSTEM

In our proposed system we have illustrated different entities in **Figure 1** data owner client, cloud service provider and cloud server.

1. **Data owner:** The data owner is the real authorized person who stores private data or business information on cloud.
2. **Cloud server:** The cloud server is covered with fog network which processes the client request and grant access to the cloud.
3. **Admin user:** The admin manages user logs, files, creates file signature, manage decoy data base or files.

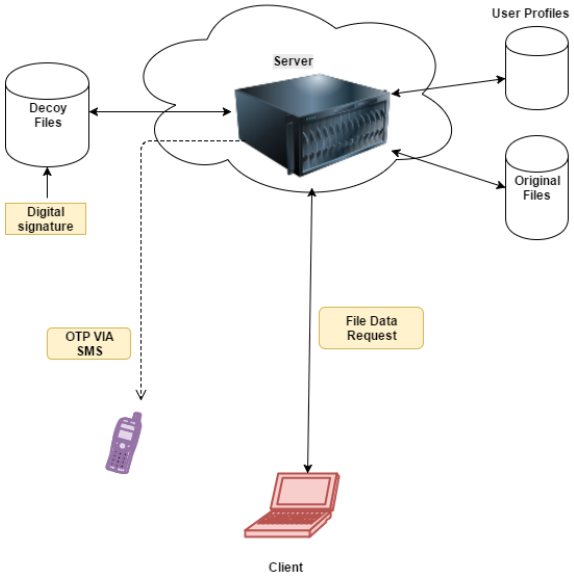

Figure 1: System Model (Proposed system)

After registration of new user, client gets the desired space on cloud and is able to perform valid operation on cloud data-base such as add new files, delete files, download files, search files, and ask for one-time password(OTP) for verification. Whenever user requests for data the request is received by cloud service provider before responding to the client request it will load user profile activity logs and apply mining technique and predict/calculate current request parameters or it will check the user patterns; if it is valid then real information is given to the user otherwise fog network launches disinformation attack and it will send fake or decoy information to user and this scenario will be immediately reported to the admin and system logs will be updated. But sometime there is a possibility of real user patterns being different and hence do not match with the system; at that time fake data is provided to real user as well, but at that time owner of data knows the system is sending decoy data as the owner is aware of the of the content or information; in this particular situation the real user can challenge the system by asking one-time password (OTP) for verification of his/her identity. OTP function is also secured with secure hash algorithm (SHA-1). This cryptographic hash function is helpful against Man In the Middle Attack (MIM), therefore it will improve the security of the system. This proposed system also maintains transparency because all the system mechanism is hidden from the user or attacker. The system admin can also perform valid operation such as, manage decoy files, create file signature and update user logs.

III. EXISTING SYSTEM

Following are the existing systems that imply fog computing.

3.1. Smart grid system: The fog computing plays an important role in smart grid system. In this system as per energy demand, availability of these devices automatically switches to alternative energies like solar or wind. The Fog collectors at the edge processes the data generated by grid devices and sensors and send control commands to the actuators. It is used to filter the data which is locally consumed and send to the higher tiers for visualization, transactional analytics and real time report data.

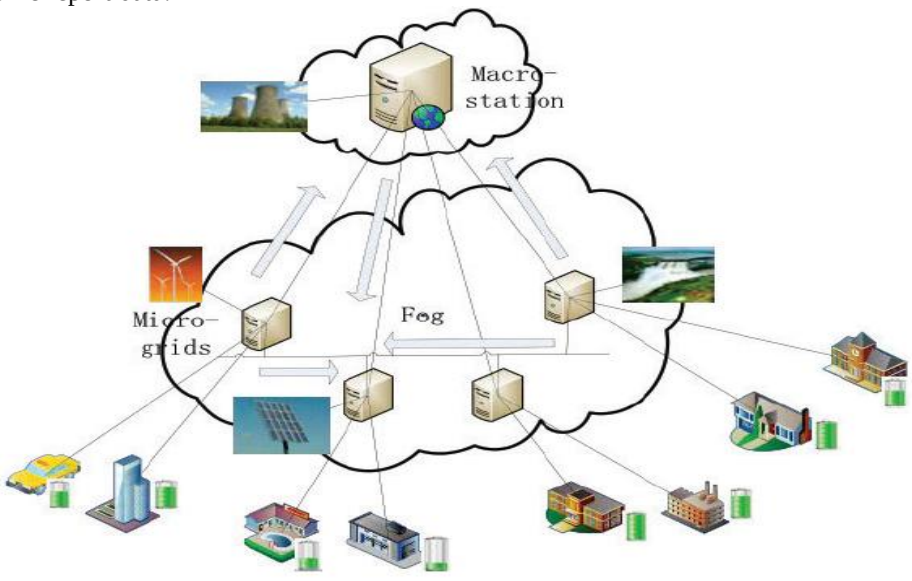

Figure 2: Fog computing in smart grid.

3.2. Smart traffic light system:

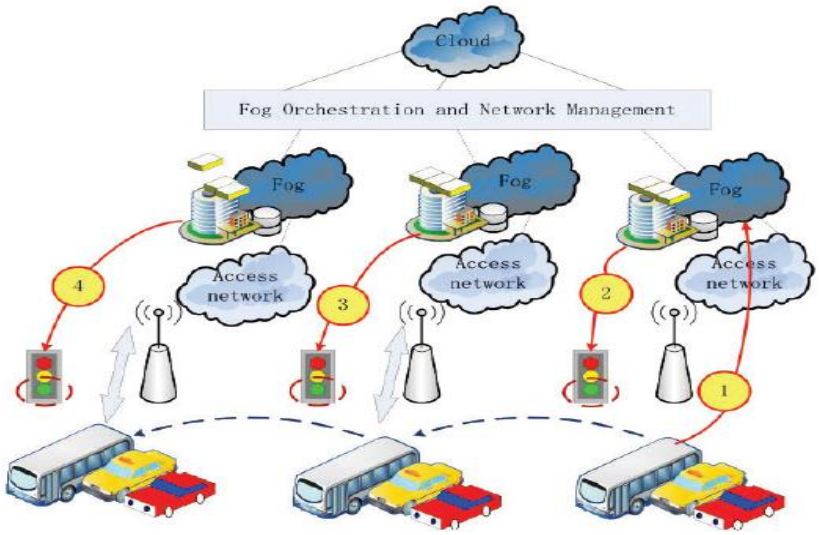

Figure 3: Fog computing in smart traffic light system.

The above system has video cameras that sense an ambulance flashing light so it can automatically change street light and open or clear lanes for the ambulance vehicle to pass through traffic. These sensors fetch the required input for the system to measure speed and distance of an oncoming vehicle. This system light turns green once sensors identify movements and turn red as traffic passes. Neighboring smart lights serving as fog devices coordinate to create green traffic wave and it will send warning signals to oncoming vehicles.

IV. SECURING CLOUD BY USING FOG NETWORK

Various methods are implied for securing data on cloud server by using different type of techniques. Sometimes this technique fails or is unsuccessful in securing user cloud data from insider attacks and sometimes other reason also come into picture such as, misconfiguration of services and other bugs in code.

4.1. User Behavior Profiling: User profiling is a well-known technique that can be applied here to how, when and how much a user accesses their information from cloud database. The system continuously monitors user behavior to check whether the pattern is normal or else abnormal access or unauthorized access to user information might be in action. Each user has a distinct profile consisting of number of times the user has accessed his files from cloud server. If there is any divergence in user behavior against the profile which is already stored in database, then it can be identified as an invalid user and attack is detected.

4.2. Decoys: Decoy information are the fake documents, trap-files, honey-files and other fake information that are uploaded by cloud system administrator on system. Fake information contains all false data which create confusion to attacker.

This technique is incorporated along with user behavior profiling. When unauthorized access is identified then disinformation attack is launched and decoy data base starts providing fake data to particular user in such a way which is completely legitimate or legal or normal. Only true owner user of data can identify when fake data is provided by cloud data base then real user can ask one-time password for verification. This secures users actual data on cloud and protects it from misuse of real data by unauthorized user.

V. COMBINING USER BEHAVIOR PROFILING AND DECOY TECHNOLOGY FOR MASQUERADE DETECTION

5.1. User Behavior Profiling The legal users of the system are familiar with the file system and the location where they are stored. Whenever there is a masquerader who gets the access of real user system illegally, is unlikely to be familiar with file structure and data or content of that file system. This fake user search is likely to be widespread or untargeted or we can say their search may not be to the point.

Basically user search behavior is profiled and developed based on some key assumption. Models are trained with a one class modeling technique i.e. one class support vector machine. It will maintain user privacy and secure the user data. All the real or normal user patterns are modeled. This model is helpful to compare and determine whether the user accessing the system is real user or a masquerade.

5.2. Decoy Technology: All the data within decoy data base are fully filled with honey -files, trap-files, fake or with bogus information which is uploaded by system admin or cloud service provider of the system.

A masquerader who is unaware with system or file system and location of the data it will try to click on decoy files. Therefore, the system is notified of unauthorized activity.

The advantages of placing decoys in a file system:

1. It will be helpful for detection of illegal or masquerade activity.
2. It will create confusion to the attacker or insider.

3. The combination of decoy technology and user behavior profiling produce strong proof or evidence of illegal or unauthorized activity of accessing the data and is also helpful to improve the accuracy of detection.

The user behavior is identified by using data access patterns. This pattern may be determined by the number of upload, download count, time of accessing data and session. All these records are maintained by the system. When any users access the system, his behavior is matched with the user behavior profile which is stored in the database. If the behavior or patterns is matching, then we can say that the user is real and accessing the data or it said to be user is illegal who has gained access to data.

VI. ALGORITHM AND EXAMPLE

6.1 Naive Bayes: Naive Bayes is a classification method or a technique based on the Bayes theorem with an assumption of independence between predictors. In other words, we can say a naive bayes classifier that assume presence of a particular feature in class is unrelated to the presence of any other feature.

Likelihood — Class Prior Probability

$$P(c \mid x) = \frac{P(x \mid c)P(c)}{P(x)}$$

Posterior Probability — Predictor Prior Probability

$$P(c \mid X) = P(x_1 \mid c) \times P(x_2 \mid c) \times \cdots \times P(x_n \mid c) \times P(c)$$

$$V_{nb} = \text{argmax}_{v_j \in V} P(v_j) \prod P(a_i|v_j) \quad \text{.....(1).}$$

$$P(a_i|v_j) = \frac{n_c + mp}{n + m} \quad \text{.....(2).}$$

where:

n = the number of training examples for which v .
nc = number of examples for which v = vj and a.
p = a priori estimate for P.
m = the equivalent sample size.

6.2 The Classifier Example:

Table 1: Classifier Example.

No.	Time-Slot	Session-Duration	Upload-Count	Download-Count	Black-list Count	Output
1.	Morning	Medium	Normal	Normal	Low	Valid
2.	Morning	Short	Normal	Normal	Low	Invalid
3.	Noon	Long	Normal	Normal	Low	Valid
4.	Evening	Long	Normal	Normal	Low	Valid
5.	Evening	Long	Normal	Normal	Medium	Valid
6.	Night	Long	Normal	Abnormal	High	Invalid
7.	Night	Long	Normal	Abnormal	Medium	Invalid
8.	Night	Long	Abnormal	Abnormal	High	Invalid
9.	Night	Long	Normal	Abnormal	High	Invalid
10.	Noon	Medium	Abnormal	Normal	High	Invalid
11.	Evening	Long	Abnormal	Normal	High	Invalid
12.	Noon	Medium	Normal	Normal	Medium	Valid
13.	Night	Long	Abnormal	Abnormal	Low	Invalid
14.	Night	Long	Abnormal	Abnormal	Low	Invalid

6.3 Training example:

We have P(Valid) = 5/14=0.357and P(Invalid) = 9/14=0.642.

Suppose we want to apply naïve Bayes classifier for (Night, Short, Normal, Abnormal, Medium).
Note there is no example of (Night, Short, Normal, Abnormal, Medium) in our dataset. Looking back at equation (2) we can see how to compute this. We need to calculate the probabilities
P (Night | Valid), P (Short | Valid), P (Normal | Valid), P (Abnormal | Valid), P (Medium | Valid).
P(Night | Invalid), P(Short | Invalid) ,P(Normal | Invalid) ,P(Abnormal | Invalid) ,P(Medium | Invalid) and multiply them by P(Yes) and P(No) respectively.

Table 2: Attributes and Values

P(Night | Valid)=0/5=0	P(Night | Invalid)=6/9=0.67
P(Short | Valid)=0/5=0	P(Short | Invalid)=1/9=0.11
P(Normal | Valid)=5/5=1	P(Normal | Invalid)=4/9=0.44
P(Abnormal | Valid)=0/5=0	P(Abnormal | Invalid)=6/9=0.67
P(Medium | Valid)=2/5=0.4	P(Medium | Invalid)=1/9=0.11

Looking at P(Night | Valid), we have 5 cases where vj = Valid , and in 0 of those cases ai = Night. So for P (Night | Valid),n = 6 and nc = 0. Note that all attributes are binary (two possible values).

We have P(Valid) = 0.428 and P(Invalid) = 0.571, so we can apply equation (2). For v = Valid, we have,

P(Valid) * P(Night | Valid) * P(Short | Valid) * P(Normal | Valid) * P(Abnormal | Valid) * P(Medium | Valid)
= 0.428 * 0 * 0 * 1 * 0 * 0.4 = 0.0

and for v = Invalid, we have,

P(Invalid) * P(Night | Invalid) * P(Short | Invalid) * P(Normal | Invalid) * P(Abnormal | Invalid) * P(Medium | Invalid)
= 0.571 * 0.07 * 0.11 * 0.44 * 0.67 * 0.11 = .0001

Since 0<0.0001, our example gets classified as Invalid
Thereby the user gets classified as INVALID USER.

6.4 Security: This system maintains integrity of data and Transparency of mechanism which works in the background and also provides security to the user's password and one-time password in database. All the user's password and one-time password are saved in database in the form of hash value by using SHA11 algorithm.

6.5 SHA-1 (Secure Hash 1) Algorithm: In cryptography Secure Hash Algorithm 1 is a hash function designed by the U.S National Security Agency. Secure hash algorithm 1 produces a 20-byte (160-bit) hash value knows as message digest.

6.6 Pseudo code that is used to encode the password by SHA-1

```
MessageDigest md = MessageDigest.getInstance("SHA-1");
byte[] passbyte;
passbyte = "abcpqrs19".getBytes("UTF-8");
passbyte = md.digest(passbyte);
```

6.7 Why SHA 1? The Secure Hash Algorithm 1 is stronger than MD5 hash Algorithm because SHA-1 generates a hash of 160-bits whereas, MD5 generates a hash of 128-bits. MD5 algorithm does not work against collisions attack whereas SHA-1 is stronger against collision attacks. MD5 algorithm can be cracked easily by attacker. On other hand SHA-1 is difficult to crack. Therefore, SHA-1 is recommended more than MD5 algorithm for hashing.

6.8 SHA-1Vs MD5:

Table 3: SHA1 vs. MD5.

SHA-1 Algorithm	MD5 Algorithm
1.SHA-1 has 160-bit hash value	1.MD5 has 128-bit hash value
2.SHA-1 support 80 rounds	2.MD5 support 64 rounds
3.Stronger against collision attacks	3. Weak against collision attacks
4. Slower than MD5 because it required 80 iterations.	4. Faster than SHA-1 because it required 64 iterations.
5. It required 2^{160} bit operation to break.	5. It required 2^{128} bit operation to break.

VII. IMPLEMENTATION

This system is divided in two panels Admin panel and User Panel.

Admin Panel:

Admin is user who has authority to manage dataset, files, load dataset, create signature on file, manage decoy database and can perform various operations.

User panel:

User is a client and there can be single or multiple users associated with username, password who store and retrieve data form cloud storage and also perform valid operation on database such as, upload files, download files, search

file, delete files and request for one-time password for verification of real user identity. In user panel the one-time password plays a very important role; if in case real user gets decoy data from cloud server, the user can request for one-time password for verification challenge.

7.1 Admin Panel: This is the system admin panel as shown in figure; here admin login with his name and password to perform further operations. After successful login admin can then perform management operations on the database.

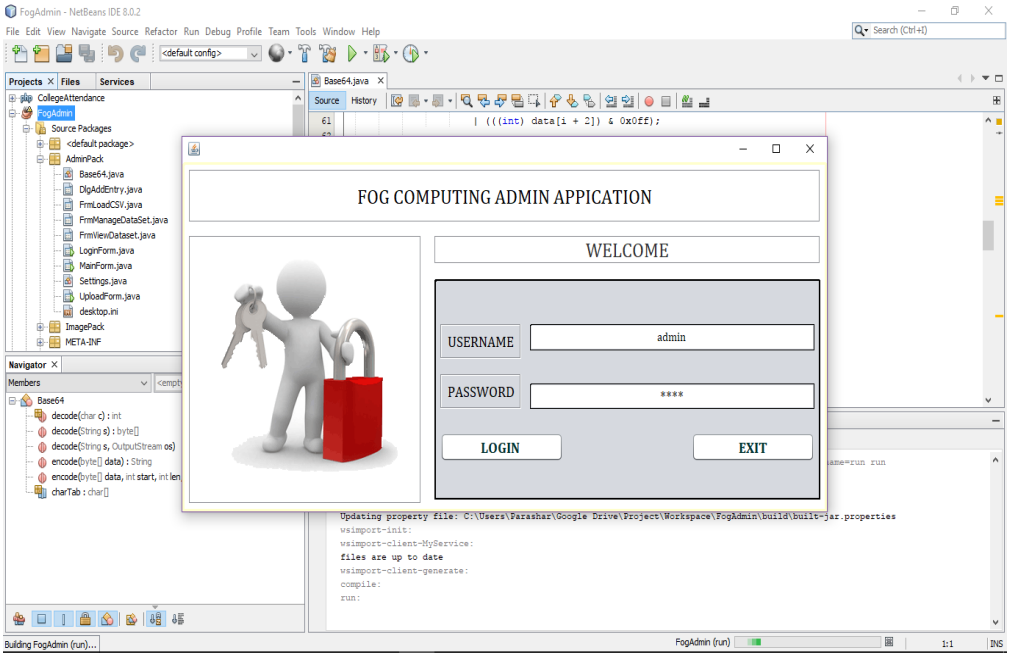

Figure 4: Admin Login Application.

After successful login the admin can manage files, dataset and load new dataset. By clicking on load data set admin can import new data from file system After clicking on manage data-set admin can create new multiple folders, upload new files and delete unwanted files. The Manage data set shows user activity such as time, duration, upload count, download count, and blacklist count as well as verify user identity whether it is normal user or invalid user.

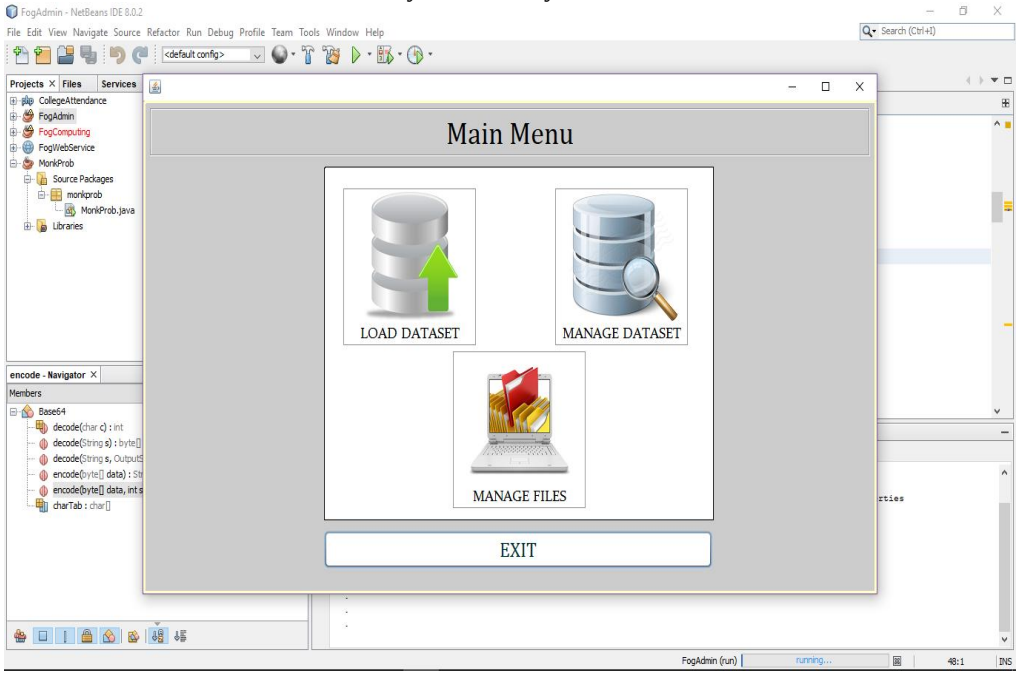

Figure 5: Admin Application.

7.2 User Panel: A new user can create new account by providing basic information and all of this information are stored in a database. Once user is successfully registered then user can login with his user name and password to access application. If user enter wrong password and user-id, then access is denied by system or user can click on forgot to set-up a new password. The following fig shows the user registration form.

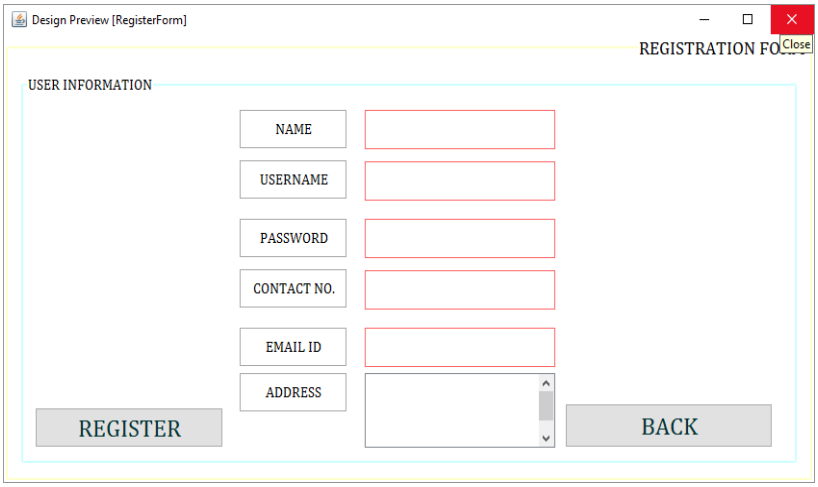

Figure 6: User Registration Form.

After successful registration of new user; the user can then login by providing valid username and password for authentication. If user is a valid-user, then user can perform valid operations on drive and access real data from cloud. If user is unauthorized then server delivers decoy data to that particular user.

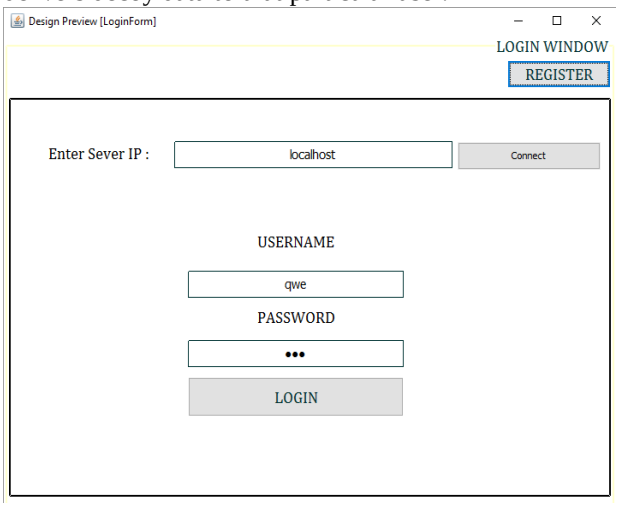

Figure 7: User Login Form.

The entire unauthorized user is black-listed and all the activity logs are stored in the database. The system admin can verify how many valid users have accessed valid data and how many in-valid users are blacklisted.

After successfully login, user can access application and perform various activities such as add files, access drive and manage logs. By clicking on access drive user can views his all files as well as user can directly search particular file by typing-in the file name, download files and delete unwanted files. User can select file and download file from cloud server. If user is valid then original data is provided by cloud server and if user is in-valid then cloud server sends decoy data to in-valid user. The user activity logs show the user activity such as previous login, data access time, upload, and download files logs. Fig shows the user application main form.

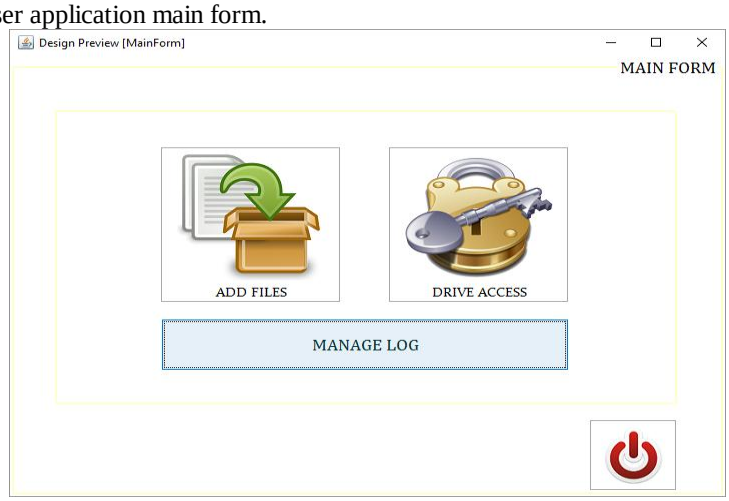

Figure 8: User Application.

All the information of admin and user client are stored in database tables. The system backend is MYSQL database which stores the user information according to tables. Database contains various tables such as files, hash table, user info, user profile etc. as show in fig. In the previous system Man-In-The-Middle attack can expose the fog computing network; but in our system we overcome this problem by providing security by the OTP function and user password in data base.

For security purpose user password and OTP are encoded by SHA-1 algorithm and then stored in the form of hash value. Therefore, the system security increases and secure from Man-In-The-Middle attack.

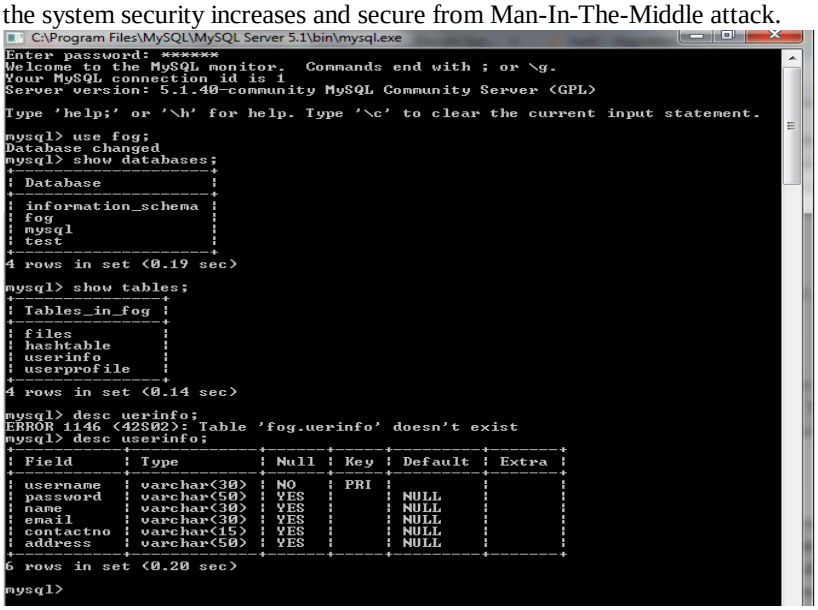

Figure 9: System Database.

VIII. ADVANTAGES AND FUTURE SCOPE

8.1. Advantages:

1. The data stored on the cloud can be stored in a secure way.
2. The system maintains data integrity.
3. The System protects against misuse of real user data.
4. It will provide security against MIM (Man In the Middle) attacks.
5. This system creates confusion for attacker by using or placing decoy files in file system.
6. It will help to detect illegal data access and provide false information to the unauthorized person.

8.2. Future Scope:

1. We can develop Android and iOS application for mobile.
   to secure cloud data.
2. Study of how attacker's behavior changes according to their knowledge about the monitoring method on the target system
3. Data can also be divided and stored on multiple clouds for extra security.
4. Hadoop framework can be used for distributed storage and processing of very large data sets.
5. High version of SHA algorithm can be use to improve system security.

IX. CONCLUSION

An application for securing original cloud data from unauthorized user and providing real or decoy data to user as based on users patterns; here user takes the advantage of the entire feature which are provided by this application. For this user simply required internet connection to establish connection with cloud data server. This system identifying user's real patterns if it is match then this system provide real data from cloud data server to user and if any unauthorized or attacker want to access cloud data then by using fog computing this system provides decoy data to unauthorized user. Incase authorized user getting decoy data if user patterns not matching then in that situation real user can ask one-time password (OTP)for verification. By using this system private and business information can be safe from third party user or hackers. All this mechanism working in background of the system therefore this system maintains transparency, maintain data integrity and create confusion for attacker by giving decoy information from decoy data base.

REFERENCES

[1] Parashar Sangle ,Rohit Ghodake, Romit Deshmukh, Akash Yadav, Prof.Jitendra Musale. "Data Security System in cloud by using Fog Computing & Data Mining" International Journal of Engineering and Computer Science ISSN: 2319-7242 Volume 5 Issue 12 Dec. 2016, Page No. 19489-19493.

[2] Ben-Salem M., and Stolfo, Angelos D. Keromytis,"Fog Computing: Mitigating Insider Data Theft Attacks in the Cloud," IEEE symposium on security and privacy workshop (SPW) 2012.

[3] Sayali Raje,Namrata Patil,Shital Mundhe,Ritika Mahajan "Cloud Security Using Fog Computing " Proceedings of IRF International Conference, 30th March-2014, Pune, India, ISBN: 978-93-82702-69-6.

[4] Ivan Stojmenovic,Sheng Wen "The Fog Computing Paradigm:Scenarios and Security Issues" proceeding of the 2014 federated conference on computer science and information systems pp. 1-8 DOI:10.15439/2014F503 ACSIS,Vol.2.

[5] Nadhiya Nazeer khan "Fog Computing: A Better Solution For IoT" International Journal of Engineering and Technical Research (IJETR) ISSN: 2321-0869, Volume-3, Issue-2, February 2015.

[6] Ben-Salem M., and Stolfo, "Decoy Document Deployment for Effective Masquerade Attack Detection," Computer Science Department, Columbia University, New York.

[7] Manreet kaur, Fog Computing Providing Data Security: A Review, International Journal of Advanced Research in Computer Science and Software Engineering.

[8] F. Bonomi, "Connected vehicles, the internet of things, and fog com- puting," in The Eighth ACM International Workshop on Vehicular Inter- Networking (VANET), Las Vegas, USA, 2011.

[9] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in Proceedings of the First Edition of the MCC Workshop on Mobile Cloud Computing, ser. MCC'12. ACM, 2012, pp. 13–16.

[10] G.Subbalakshmi, K. Ramesh, M. Chinna Rao"Decision Support in Heart Disease Prediction using Naïve Bayes" Indian Journal of Computer Science and Engineering(IJCSE).

[11] Eric Meisner "NaïveBayesClassifierExample" Nov 22,2003

