Frontiers in Health Informatics
ISSN-Online: 2676-7104
www.healthinformaticsjournal.com
2024; Vol 13: Issue 7
Open Access

Preserving Privacy In Chronic Disease Big Data Manipulation For Fog To Minimize Latency And Enhance Security Using Decoy Implementation

Mr. Yogesh R. Chikane[1*], Dr. Rashmi Soni[2]
[1]Research Scholar, Department of Computer Science and Engineering Oriental University, Indore
[2]Research Supervisor and Associate Professor, Department of Computer Science and Engineering Oriental University, Indore

Cite this paper as: Mr. Yogesh R. Chikane, Dr. Rashmi Soni (2024). Preserving Privacy In Chronic Disease Big Data Manipulation For Fog To Minimize Latency And Enhance Security Using Decoy Implementation. *Frontiers in Health Informatics*, *13 (7) 970-984*

ABSTRACT

The present disease stage is provided in this study, and sensitive data is stored securely on the cloud. The patient's chronic disease symptoms will serve as the input for the tests that will be suggested. Heart disease, asthma, diabetes mellitus, cancer, and many more are examples of chronic diseases. The test results serve as input parameters, and those parameters are then transferred to database ideals. The patient will receive a report detailing the patient's chronic disease stage after the mapping process is complete, and the information will be securely kept on the cloud. Fog handles all the data processing, while the cloud just stores the results. Decoy files are another way that fog computing is utilised to provide security. In order to fool attackers, decoy files include false or erroneous information. Unauthorised individuals receive it. Encryption is employed to reduce the impact of data theft attacks. The patient will receive a report in the mail outlining their chronic disease status as well as treatment options in various cities. For the purpose of anytime access, the report is kept on the cloud. For security reasons, the system only stores fake files. The fake file is used in the event that any unauthorised access attempts to retrieve data. The use of fog computing for security & cloud computing for storage has allowed us to maintain the privacy of patients' medical data.

KEYWORDS: *Big Data, Decoy Files, Cloud Computing, Security, Fog Computing, Medical Data*

1. INTRODUCTION

In the age of big data, the convergence of extensive data collection and sophisticated analysis techniques has driven remarkable advancements across various sectors. However, this surge in data manipulation raises significant concerns regarding privacy and security. Companies are starting to rely more and more on fog computing, a decentralized model that brings cloud computing capabilities to the network's periphery. This paradigm is essential for protecting sensitive data & maintaining latency to a minimum.

Fog computing, by processing data closer to the source, offers a powerful solution to reduce latency and improve the efficiency of data handling. Yet, this localized data processing also amplifies the need for robust privacy-preserving mechanisms. The challenge lies in implementing strategies that safeguard personal information without compromising the performance benefits that fog computing provides.

One promising approach to address these concerns is the use of decoy implementation in big data manipulation. By incorporating decoys false data or misleading signals into the data processing framework, organizations can obscure the true nature of sensitive information, thereby enhancing security and privacy. This technique not only complicates potential attacks by creating ambiguity but also helps to protect against unauthorized data access and inference attacks.

This introduction sets the stage for exploring the integration of decoy implementation within fog computing environments. It will delve into the methods for preserving privacy in big data manipulation, discuss how decoy techniques can minimize latency, and highlight their role in enhancing overall security. As we navigate this intersection of privacy, performance, and security, the insights gained will be pivotal in developing more resilient and efficient data processing frameworks for the future.

2. RESEARCH METHODOLOGY

Medical data is expanding at a dizzying rate these days. The primary goal of this study is to safeguard patients' sensitive data stored in the cloud while also delivering updates on their illness progression. We will create a mapping between the patient's textual input & ideal values already saved in the database. Fog handles all the data processing, while the cloud just stores the results. The advantages of cloud computing in terms of processing power, data storage, & connectivity are brought closer to the consumer. Its primary function is to reduce idleness. Concerns about data theft and other security breaches are plaguing the medical care cloud. The use of encryption & decryption, in addition to the additional security given by inserting fake files in the fog, helps to diminish the value of these data theft attacks. Once we identify unauthorised access, we provide the attacker with a decoy file that contains inaccurate information. The AES symmetric key cryptography algorithm is used for both encryption and decoding. Finally, patients are given accurate disease diagnoses along with treatment cost comparisonsacross various facilities.

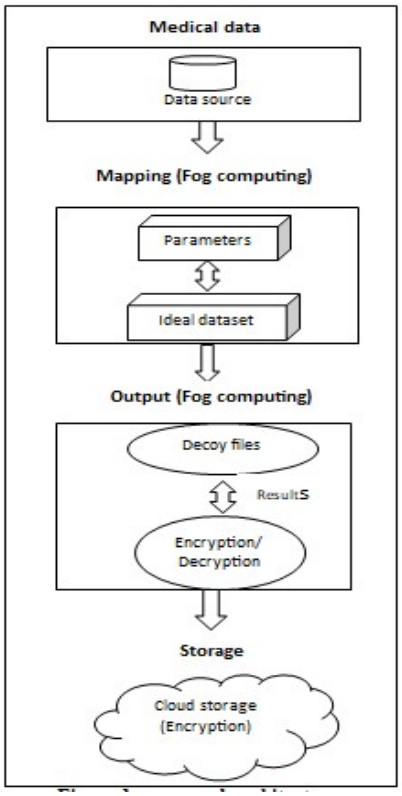


Figure 1: proposed architecture

- **Medical Data:** Here, we collect information about the patient's symptoms and provide them with test suggestions for conditions including diabetes & heart disease. The following tests are recommended: ECG, blood, AIC, Holter Monitor, etc.

- **Mapping:** The next step is to use fog to map the parameters extracted from the test results to the ideal dataset. By using certain settings, patients can determine the stage of their chronic disease by mapping, which is done using optimal parameters that inform them the result of heart disease or diabetes.
- **Output:** At this point, the patient receives a report (the chronic disease stage), together with certain precautions and information on hospitals in various cities, through registered mail. The report triggers the automatic creation of a decoy file, which includes a random generation mechanism and, for security reasons, encryption.
- **Storage:** The report's encrypted files are saved on the cloud, so they may be accessed whenever needed. One storage option is the cloud, and our fog, which is a laptop, also served to store the encrypted & fake files.

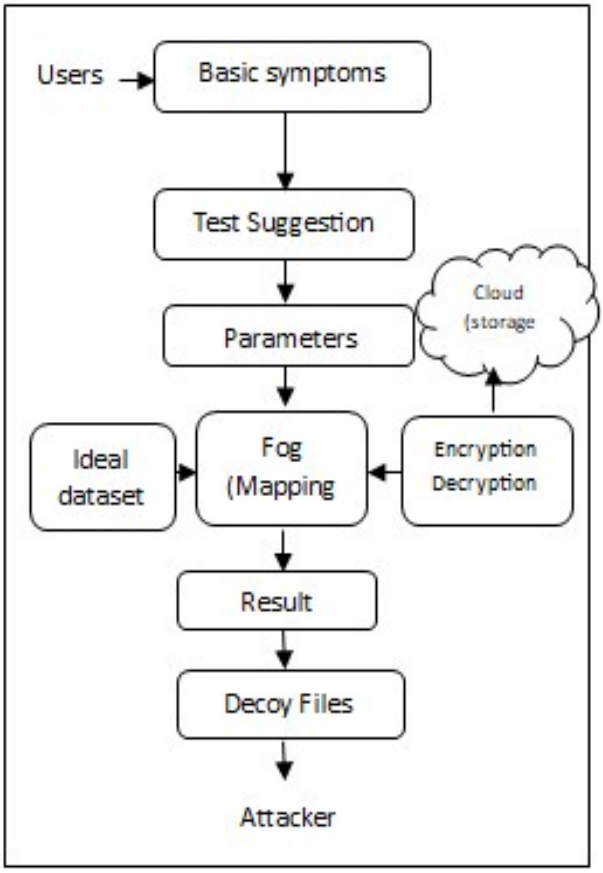


Figure 2: Data flow diagram

- **User:** After creating an account, users can log in. He or she must also provide personal information during the registration process. The "Forgot Password?" button allows the user to recover their forgotten password. Users will receive an email with their password when they click the "Forgot Password" button.
- **Basic symptoms:** Once the user has successfully registered, they are asked to choose the most basic symptoms they are experiencing. Chest discomfort or pressure, weakness, pain in the jaw, neck, or throat, irregular heartbeats, excessive weariness, etc. are the most basic signs of heart disease. Nausea, extreme fatigue, increased hunger, blurred vision, increased thrust, patches of dark skin, etc. are the primary symptoms for selecting diabetes in a patient.
- **Test suggestion:** We advise the patient to get tests like blood work, Holter monitor, electrocardiogram, AIC, etc. We will fill out the next set of parameters after the patient completes it from the hospital.
- **Parameters:** The results of the tests will reveal things like cholesterol levels, blood sugar levels (both at rest and when fasting), blood pressure, the presence or absence of heart infections, the history of heart attacks, and the

presence or absence of diabetes. We get your haemoglobin level, blood sugar, and urine tested. Previously saved the perfect dataset that maps to these parameters.

- **Fog:** The fog is used for mapping and the generation of decoy files. Additionally, the patient receives reports from fog. Our laptop acts like a fog, collecting data from smart devices, processing it, and storing the results in the cloud.
- **Result:** This study's findings include the chronic disease stage, as well as certain safety measures and hospitals in various places that offer the best treatment options.
- **Decoy files:** Decoy files are intentionally misleading because they include false information. To make these, we use a random generation process to come up with values that aren't the same as the original parameter. The system would save decoy files and make them accessible to unauthorised users.
- **Encryption/Decryption:** We encrypt the report for security reasons & decrypt it when the patient asks for it. To transform plaintext into figure text, one must use encryption; to convert figure text back into plaintext, one must use unscrambling. We import the AES algorithm & utilise it for encryption and decryption.
- **Cloud:** Stored securely on the cloud, reports can be accessed at any time.

3. FOG COMPUTING

The term "fog computing" refers to a decentralized computing architecture or method in which data sources and other data centers are physically separated. One model that addresses user demands at the edge networks is fog computing. Devices including routers, gateways, bridges, and hubs often carry out networking-related tasks at the fog layer. Theoretically, these gadgets may do processing and networking at the same time, according to the researchers. Even though these devices don't have as much power as cloud servers, their dispersed location and lack of central authority allow them to cover a large region reliably. In fog computing, the devices are located closer to the end users rather than on distant servers in the cloud.

Advantages Of Fog Computing

Fog computing is an important distributed paradigm because of its location among Cloud computing & IoT. As such, it mediates communication between the Internet of Things (IoT), cloud computing, and edge computing. This strategic positioning not only serves as a distinguishing characteristic, but also brings about other benefits that should be recognized. Here are a few important advantages:

1. **Reduced Latency**: For real-time applications like telemedicine, telesurgery, and autonomous cars, fog computing is the way to go since it cuts down on latency by processing data closer to the source.
2. **Efficient Network Utilization**: In order to improve network efficiency & alleviate congestion, fog computing can decrease the amount of data that has to be transported to the cloud.
3. **Contextual Awareness**: The requirements of customers are carefully consider throughout the creation of the Fog infrastructure. On the path from the cloud to the internet of things, this allows for granular management, storage, communication, & processing power distribution. As a consequence, apps that are perfectly suited to each client's requirements are developed.
4. **Operational Resilience**: The Fog architecture allows for the sharing of processing, storage, communication, & control capabilities, and it lies on the spectrum between the Cloud & IoT. To improve operational resilience & fault tolerance, fog nodes can operate independently of the core Cloud layer.
5. **Improved Privacy and Security**: Because fog nodes may do data processing locally, there is less need to transmit sensitive information over the network, which improves privacy & security.

Implementation Details of Fog Computing

There are subtle differences between fog computing & edge computing, although the words are sometimes used interchangeably. In contrast to Edge Computing, which focuses on nodes in close proximity to IoT devices, Fog

Computing incorporates resources from the end device all the way to the cloud. Devices such as wireless routers & machine-to-machine gateways are examples of fog computing nodes (FCN) that add a computer layer to an existing system. These nodes are vital because they process data locally and store it before sending it to the cloud.

❖ **Implementation Architecture**:

The three levels that make up a fog computing architecture are as follows:

1. **Thing Layer**: IoT equipment such as sensors, mobile phones, or smart cars are located at the lowest level, commonly referred to as the edge layer. Physical properties (like vibration or pressure), digital content (like video feeds or system logs), and environmental variables (like temperature or humidity) are all generated by the devices that comprise this layer. Wireless technologies such as Wi-Fi, Bluetooth, Zigbee, and cellular networks are used to connect devices to the network. Furthermore, a few gadgets might need to be hardwired.
2. **Fog Layer**: An integral part of any fog computing system, the fog node is where everything starts. Nodes in a fog network might be either physical devices like gateways, switches, routers, or servers, or virtual ones like cloudlets, virtual computers, or virtualized switches. Important computer resources are supplied by these nodes, which are in turn coupled in complex ways to smart end-devices or access networks. The FCNs are diverse in character, regardless of whether they exist in the real world or not. The diversity inside FCNs facilitates communication between the FCN and end-device using non-IP based access techniques or enables the support of devices operating at numerous protocol levels.
3. **Cloud Layer**: At the very top of the stack you'll find devices that supply massive amounts of storage & powerful servers. Data storage and computation analysis are carried out by this layer.

❖ **Request Handling**:

In order to enable devices that operate at different protocol levels and have different access technologies, Fog Computing takes use of the heterogeneous nature of FCNs in its decentralized architecture. To make the most of Fog Computing resources, the Service Orchestration Layer may adapt to changing needs by allocating them dynamically according to user specifications.

The Fog Orchestrator painstakingly matches the rules provided by each node with the end-user requests that include predetermined policy criteria, such load balancing and Quality of Service (QoS). As a further step, it provides a prioritized list of nodes that have been evaluated against the policy criteria. This choice takes availability into account, making sure it's in perfect harmony with what the end user needs. Requests that are time-sensitive and need low latency are handled locally by the Fog node. This includes things like altering the temperature depending on data from nearby sensors or real-time danger identification from security cameras. On the other hand, sending the request to the cloud could be the better option if it's resource-intensive but not time-bound.

This dynamic method of request management optimizes resource consumption, reduces latency, & boosts the network's overall performance. With its intelligent orchestration & targeted processing, the Fog Computing architecture revolutionizes network operations by making them more efficient & responsive.

❖ **Data Preprocessing and Contextualization**:

At the network's periphery, close to the data-generating devices, data preprocessing entails gathering, analyzing, and interpreting data. After data normalization, processing can continue with or without sliding windows; it all depends on the devices and their utilization cases.

Before sending data to the cloud, it must first be reduced. There are two types of edge data reduction: reversible and nonreversible.

1. **Reversible**: With this method, data may be decreased while still being able to be reproduced from their reduced forms. These methods include reducing data at the edge, sending it over the network, and then either immediately using machine learning (ML) on the reduced data or first reproducing the original data on the cloud.
2. **Nonreversible**: When data reduction is complete, there is no way to restore the original data, so the process is considered nonreversible.

In Fog Computing, "contextualization" is making sense of and making use of data in relation to its immediate environment, including its time, place, and source device. Fog computing is able to deliver contextually aware, tailored services because it understands the situation. In a smart home setting, for instance, the fog node may regulate the temperature depending on factors like the time of day, whether or not someone is inside, and the weather outside.

4. CLOUD COMPUTING

Cloud computing is the process of providing computer resources to users on an as-needed basis. A lot of data storage & processing power is available to us through online apps. This service is pay-as-you-go. Anyone may rent storage and apps from a cloud provider; they don't need to own any data centers or computer hardware. By utilizing cloud computing services and only paying for the resources actually used, we may sidestep the hassle of owning & maintaining infrastructure.

Providers of cloud computing services can reap the rewards of economies of scale when they serve a large number of clients at once. Infrastructure, platforms, & software applications are the many kinds of cloud services that fall under the umbrella of cloud deployment methods.

In the cloud, there are primarily three models:

❖ **Infrastructure as a Service (IaaS)**

Infrastructure as a service provides resources for computing, storage, networking, & virtualization as needed. In IaaS, the provider is in charge of the physical hardware and networks, while the software—including OSes, middleware, data, and apps—is purchased & handled by the clients.

❖ **Platform as a Service (PaaS)**

PaaS facilitates the provisioning, testing, deployment, & administration of cloud applications by delivering and managing the necessary hardware & software resources. As part of their PaaS, most providers also include development tools, cloud databases, or middleware.

❖ **Software as a Service (SaaS)**

SaaS allows users to access & utilize an entire application stack through the cloud. Many SaaS offerings are pre-built, user-friendly apps that the supplier handles all aspects of administration & maintenance.

❖ **Server less computing**

Function as a Service is another name for serverless computing in cloud service architectures. The ability to construct applications as straightforward, event-triggered functions free from the burden of infrastructure management & scaling is offered by this innovative cloud service paradigm.

Cloud Computing Deployment Models

Cloud computing can be implemented in various ways, some of which are as follows:

Private cloud

Private cloud services are provided to internal users by a company's data centre. A private cloud allows a company to construct and manage its own cloud infrastructure. This architecture provides the flexibility & ease of cloud computing with the oversight, command, and safety often associated with on-premises data centres. The IT department has the option to employ chargeback to charge internal users for services. Software like VMware & Open Stack are illustrations

of private cloud vendors & technologies.

Public cloud

The public cloud model relies on the internet-based delivery of cloud services by an independent third party, or CSP. Many public cloud services do provide longer-term commitments, but most are sold on an as-needed basis, usually by the minute or hour. Consumers are solely charged for the amount of storage space, computing power, or bandwidth that they really use. Notable public cloud service providers include Amazon Web Services (AWS), Google Cloud Platform (GCP), IBM Watson, Microsoft Azure, Oracle, & Tencent Cloud.

Hybrid cloud

A hybrid cloud integrates both public & private cloud services, with automation and orchestration between the two, in addition to an on-premises private cloud. The private cloud is ideal for mission-critical or sensitive application runs, whereas the public cloud is great for companies that experience surges in demand or workloads. The idea behind a hybrid cloud is to combine thebest features of public and private cloud computing to build an automated, scalable system that can handle mission-critical data without letting go of control.

Multi-cloud

The usage of numerous IaaS providers, or a multi-cloud approach, is becoming more popular among organizations. This enables apps to run in parallel across several cloud providers or to move between various cloud providers.

Companies implement multi-cloud strategies for a variety of reasons, such as to reduce the likelihood of cloud service interruptions or to take advantage of a provider's more favorable pricing terms. It also aids businesses in avoiding vendor lock-in by allowing them to easily swap providers if necessary.

As a result of variations in APIs and services offered by cloud providers, developing applications for multi-cloud deployments can be difficult. As more and more cloud providers strive to standardize and merge their APIs & services, multi-cloud deployments should grow easier. Open Cloud Computing Interface is one industry effort that tries to facilitate interoperability or multi-cloud deployments.

Community cloud

An organization or groups of organizations may work together in a community cloud to address shared goals, policies, security needs, & regulatory issues. These groups or an outside company administer an on-premises or cloud-based community cloud.

5. ARCHITECTURE OF FOG COMPUTING

The privacy-preserving framework was created to tackle the shortcomings of previous privacy-preserving approaches & offer a practical and scalable answer to the issue of privacy preservation in fog & cloud environments. Data collection, storage, processing, & analysis were all aspects of the larger structure. The following is a description of the framework's components: The data collection component was in charge of gathering Chronical large data from various sources & storing it safely in a database. It was imperative that the data storage component provide the security & scalability of the stored data. Preparing the collected data for analysis was the responsibility of the data processing component. In order to find ways to improve different applications, the data analysis part looked at the processed data and came up with ideas. By moving the cloud model to the network's periphery, a distributed computing architecture known as "fog computing" may process data in real time with minimal latency. The following figure illustrates the fog computing architecture:

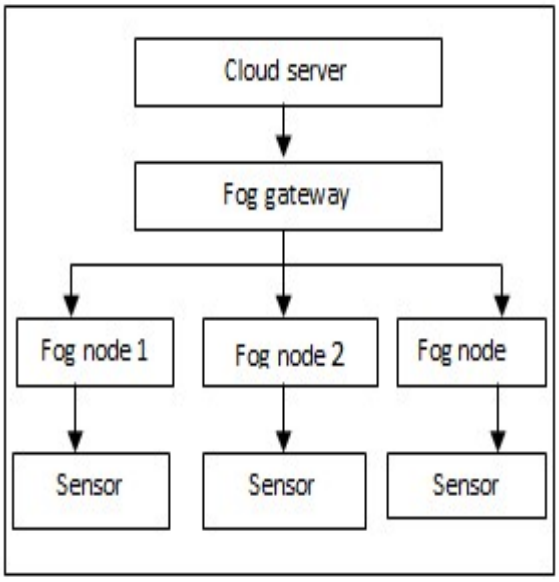


Figure 3: Architecture of fog computing

The fog gateway connects the cloud server to the fog nodes in this design. The fog nodes are connected to the sensors so that they can collect data. The data is processed in real-time by the fog nodes, who subsequently upload it to the cloud for further storage & analysis. When planning a system to secure persistent large data in the cloud and fog, it's possible to incorporate the following features:

Data processing and real-time analysis are handled by the fog nodes. After that, it's on to a cloud server for future study and preservation. The processed data is safely stored in the database, ensuring that the data remains private.

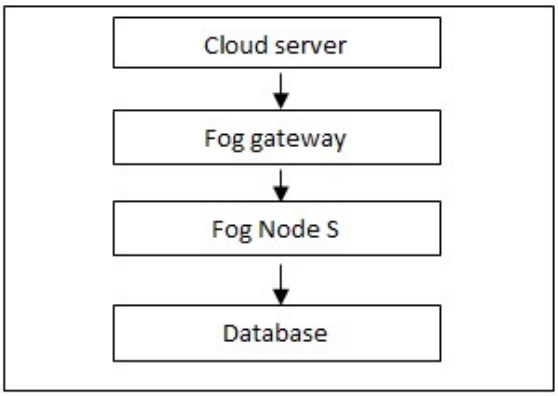


Figure 4: System Architecture

6. DECOY FILE IMPLEMENTATION

Data privacy may be maintained through the use of a decoy file implementation. Decoy files are intentionally misleading because they mimic the appearance of real data while really containing fake information. An example of a decoy file in action is shown in the figure below:

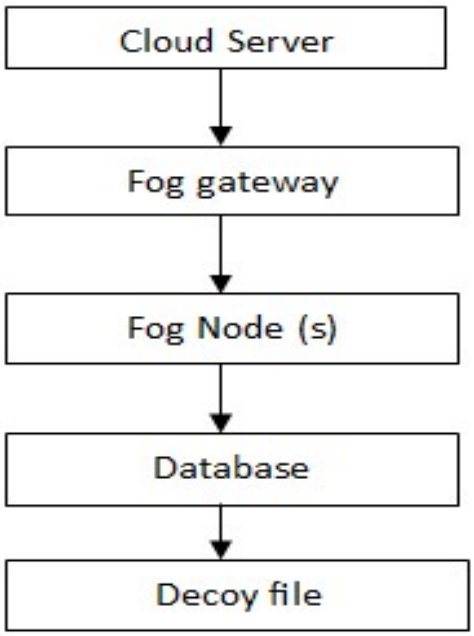


Figure 5: Decoy file implementation

This technique uses the real data to store a fake file in the database. Although it appears to be real data, the decoy file actually contains false information.Because of this, data privacy is maintained and unauthorized users are unable to access the actual data. Only authorized individuals can view the decoy file; the actual data is kept in a different, safe place. To keep Chronical6+ huge data private in cloud and fog settings, analysis for decoy file deployment is crucial.

Decoy File Importance

Often referred to as "honeypot files," "decoy files" are completely fictional documents used to deceive hackers into divulging sensitive information. These files may appear to be data files, but they do not actually hold any important information. Decoy files are used to divert an attacker's attention and make it harder for them to identify & access real important data files. Since fog and cloud environments retain a great deal of sensitive data, they are more susceptible to cyber assaults; thus, the deployment of decoy files is crucial in these settings.

Analysis of Decoy File Implementation

There are numerous steps involved in implementing decoy files, including creating the files, distributing them around the fog & cloud environment, and monitoring them for attempted unauthorized access. Each stage is explored in depth here:

1. Creating Decoy Files: Decoy files should closely resemble data files and be kept in the same folders. In terms of name, size, & format, the decoy files should closely resemble the real data files.
2. Distributing Decoy Files: It is recommended to disperse decoy files over the fog & cloud environment to impede attackers' ability to discern legitimate data files. To make decoy files less noticeable, they should be mixed in with actual data files and delivered randomly.
3. Monitoring Decoy Files: It is important to verify decoy files for any attempts at unauthorized access. Several instruments, including firewalls, intrusion detection systems, & log analyzers, are available for use in monitoring. In order to detect cyberattacks in real-time, monitoring is crucial for identifying & tracking possible attackers.

Advantages of Decoy Files

Among the many advantages of using decoy files are:

1. Enhanced Safety: Decoy files can make fog and cloud settings safer by preventing attackers from focusing on legitimate data files and making it harder for them to distinguish between the two.

2. Cyber Attack Early Detection: Decoy files can aid in the real-time detection of cyber-attacks, enabling a prompt reaction & lowering the danger of data breaches.
3. Decreased Probability of Data Breaches: The use of decoy files can decrease the probability of data breaches by making it more difficult for attackers to access real sensitive files.

Using decoy files is an effective way to increase security, detect cyberattacks in real-time, &decrease the probability of data breaches. Decoy file implementation entails creating decoy files, dispersing them about the environment, and keeping an eye on them for intrusion attempts.

Decoy File Implementation using Mathematical Procedures

If you want to utilize decoy files, you have to make a bunch of files that are identical copies of the real data files and put them in the same folders. The decoy files ought to be named, sized, & formatted in a way that closely resembles the real data files.

The following steps may be used to describe this in mathematics:

Assume that F_real contains the actual data files & F_decoy contains the mock up files. In this case, we'll use N for the total number of files in F_real and M for the specified amount of decoy files.Make a copy of each file f in F_real called f' in F_decoy. Make sure the two files have the same name, size, and format. Make a random number between M and N more decoy files with the same size, format, and name as the actual data files in F_real. The directories where the real data files are stored in F_real should also contain the decoy files. To further disguise them, the decoy files should be dispersed randomly around the cloud and fog environment with the real data files.

The following steps may be used to describe this in mathematics:

Assume that the fog and clouds surround D, the collection of directories. Choose a random selection of files from F_real and F_decoy, and save them in directory d in D. Attackers will have a hard time identifying real data files if you randomly mix decoy files & real data files within each directory. Lastly, there are a number of tools available, including firewalls, intrusion detection systems, or log analyzers, that can be used to monitor decoy files for attempts at unwanted access.

The following steps may be used to describe this in mathematics:

In a fog & cloud setting, let A represent the collection of file access attempts. You should verify if the file being accessed is a fake or actual data file for every attempt in A. Do nothing more than record the attempt to access the file if it turns out to be a decoy file. Check the access attempt's authorization status if the file in question is a legitimate data file. Keep track of the access attempt & proceed with no further steps if it is approved. Put a stop to the unauthorized access attempt and inform the proper authorities if necessary.

Algorithms

1. Algorithms utilized by the framework overview: Data anonymization, access control, & encryption methods were all a part of the system. The confidentiality of Chronical's huge data, which was processed and stored in the cloud and fog, was ensured by these techniques.
2. The algorithms that protect users' privacy are described below: Data encryption technology was employed to convert Chronicle's massive data set into a format that could only be viewed by authorized persons, therefore guaranteeing the data set's security. Also, in order to limit who may access sensitive information and when, access control measures were used. The goal of using data anonymization technology was to make it more difficult for unwanted parties to obtain sensitive data by removing identifying information.

7. EVALUATION

The software development lifecycle would be incomplete without testing, which should begin as soon as feasible and be integrated into the requirements gathering process. There is a lifecycle that includes testing. The software development lifecycle consists of the following steps: identifying a need, creating code to address that need, and finally,

testing the code to determine if it satisfies the stakeholders, who include owners, users, or everyone else with a stake in the software's functionality. As a result, we conduct tests according to the requirements of the stakeholders and launch them prior to the product's release to the public. We don't want to squander time on something that won't function, so ideally before then.

- Tests represent requirements. Your tests should be drawn from and tied to those requirements, whether you write usage scenarios in a giant thick document or hang user stories on the wall. As previously mentioned, creating tests provides an excellent opportunity to go over the specifications.
- We're not done till the tests pass. Successful completion of tests is the only meaningful indicator of progress.

8. EXPERIMENTAL RESULTS

User Registration: This figure shows the registration window view of project. User can register on the system. While registering user must enter their name, username, mobile number, e-mail id, password. "Registration Successful" message is displayed on the screen after registration.

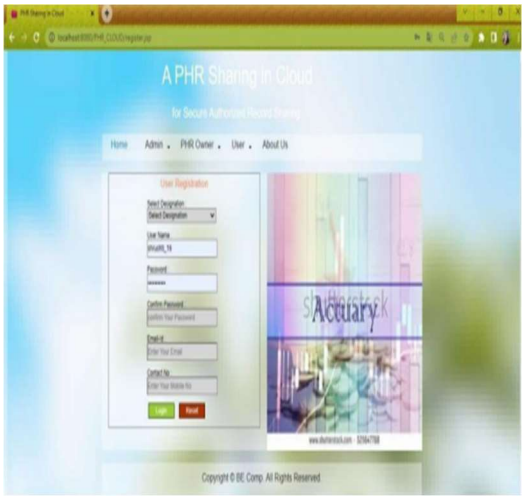


Figure 6: User Registration

User Login: This figure shows the user login window view of project. User can login on the system. While login user must enter correct user name and password. If a new user accesses that page then we have provided them with a new user registration option on the same page.

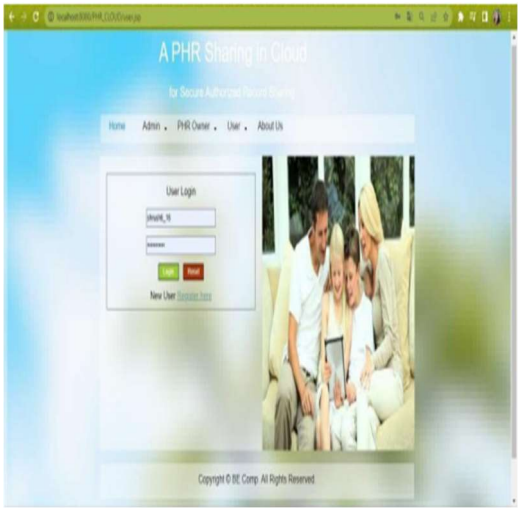


Figure 7: User Login

Admin Login: This figure shows the admin login window view of project. admin can login on the system. While login admin must enter correct user name and password.

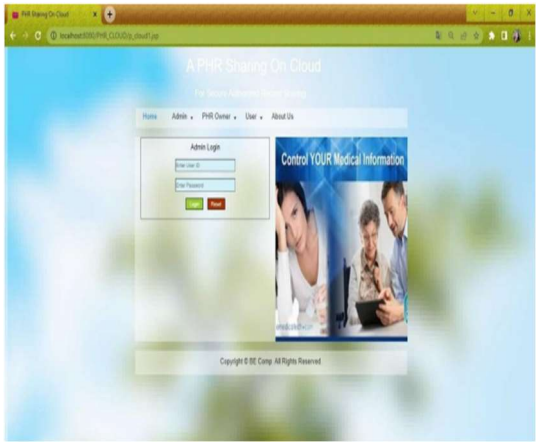


Figure 8: Admin Login

Admin View: Below figure shows the Admin window view of the project. Admin can display this view after successful login. Here the administrator can activate and deactivate the user and also maintain the system data.

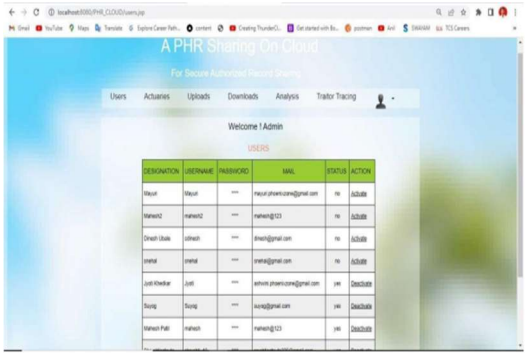


Figure 9: Admin Page

Database: Below figure shows the Database view of the project. In database user registration, Patient record, Patient registration, Actuary registration, symptoms data are stored in structured manner.

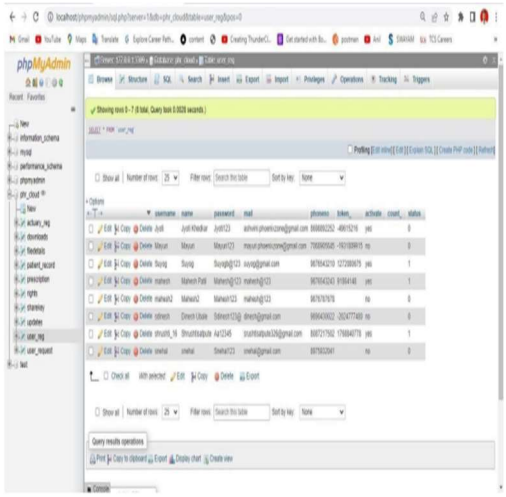


Figure 10: Server Side Database

Healthcare Dataset

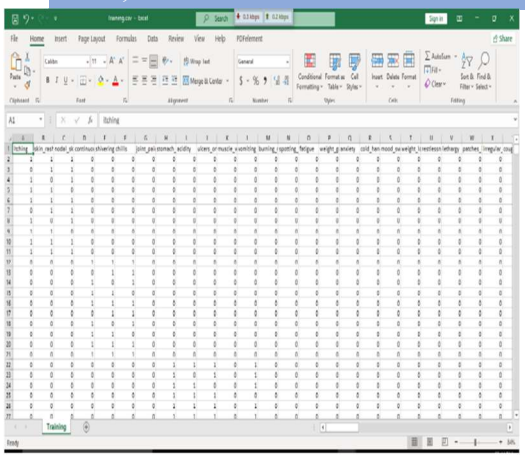
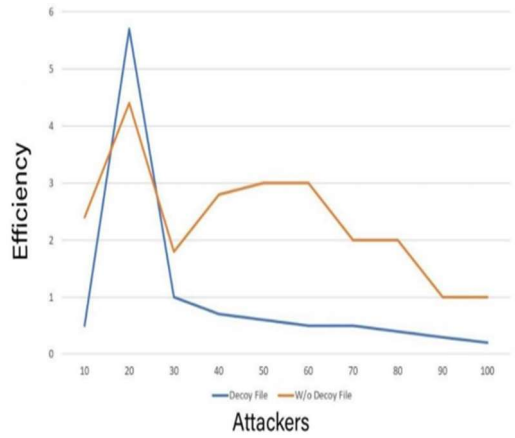


Figure 11: Healthcare Dataset

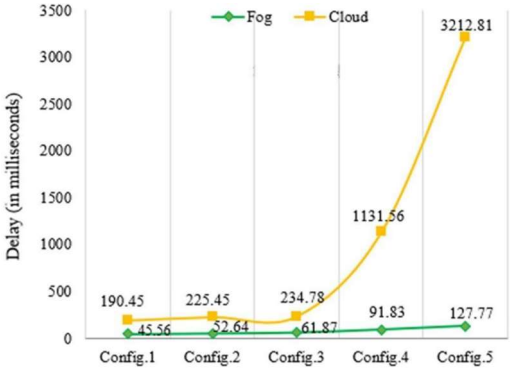
Security graph



Graph 1: Security graph

Detect threats faster and decrease attacker dwell time. By deploying and constantly monitoring decoy resources, security teams can more quickly and efficiently identify attackers in their environments than would likely otherwise be possible.

Latency Graph



Graph 2: Latency Graph

In above latency graph we can see that, time delay between fog computing and cloud. Fog Computing is more robust and efficient than cloud where we can see that cloud takes more time to configure the document.

9. PERFORMANCE EVALUATION

Performance metrics: Various metrics were utilized to evaluate the framework's performance, which include those for compute efficiency, scalability, & privacy preservation. Findings from the assessment of performance: The performance evaluation also revealed that the framework was efficient & scalable, so it could be used on a big scale. The results showed that compared to previous privacy-preserving methods, the framework outperformed them in terms of both computing efficiency & privacy protection.

Instead of depending entirely on centralized cloud servers, the computation & storage are moved closer to the network's edge in fog computing, a distributed computing paradigm. One possible use case is in healthcare settings, where there is a need to manage large volumes of sensitive data while protecting individual privacy.

The utilization of decoy files is one method for protecting the confidentiality of medical big data. Decoy files are intentionally misleading because they contain irrelevant or erroneous information while appearing to be legitimate files. To make it harder for attackers to discover the actual facts, they are utilized to muddle their thinking.

The following algorithm can serve as the basis for a mathematical model that generates decoy files:

1. Select a collection of actual data files that are typical of the information that requires security.
2. Extract statistical parameters like mean, standard deviation, & correlation coefficients from each real data file by analyzing the data.
3. Make a collection of fake files that mimic the actual data files in terms of their statistical properties. One approach or variation auto encoder (VAE) to build the decoy files is to use a generative model, including a generative adversarial network (GAN).
4. Enhance the decoy files with noise for a more lifelike appearance. Randomly perturbing the statistical features is one way to achieve this.
5. Use a distributed storage solution, like a fog computing environment, to keep both the real data files & decoy files.

The decoy files will be easily detectable by an attacker attempting to access the data, but they will be unable to differentiate them from the actual data files. Consequently, they will be unable to derive any useful insights from the data. Keep in mind that high-quality statistical features utilized to create the decoy files are crucial to the decoy file approach's efficacy. On top of that, the strategy might not work against complex assaults that can detect statistical discrepancies between the actual data & mock records. For that reason, you should incorporate this method with others, including encryption & access control, into a comprehensive security plan.

10. CONCLUSION

This study concluded the securing sensitive patient data stored in the cloud and presenting the patient's current disease stage is the main goal of this study. There is no way for sensitive data to fall into the wrong hands because it is protected. Both the provider & patient benefit from the time savings that result from the prompt delivery of services. Its primary function is to reduce idleness. Concerns about data theft and other security breaches are plaguing the medical care cloud. The utilisation of encryption & decryption, in addition to the additional security given by inserting fake files in the fog, helps to diminish the value of these data theft attacks. Once we identify unauthorised access, we provide the attacker with a decoy file that contains inaccurate information. AES symmetric key cryptography algorithm is used for both encryption & decoding.

REFERENCES

[1]. Adriana Alexandru, Dora Coardos. “Big Data In Healthcare And Medical Applications In Romania". June 2016
[2]. AkhileshVishwanath, RamyaPeruri, Jing (Selena) He.“Security in Fog Computing Through Encryption”. MECS, 2016, 5, 28-36.
[3]. AkhileshVishwanath,RamyaPeruri,Jing (Selena) He "Security In Fog Computing Through Encryption" IEEE(2016)

[4]. Al Hamid, H. A., Rahman, S. M. M., Hossain, M. S., Almogren, A., &Alamri, A. A security model for preserving the privacy of medical big data in a healthcare cloud using a fog computing facility with pairing-based cryptography. IEEE Access, 5, 22313-22328. (2017).

[5]. Alzoubi, Y. I., Osmanaj, V. H., Jaradat, A., & Al-Ahmad, A. Fog computing security and privacy for the Internet of Thing applications: State-of-the-art. Security and Privacy, 4(2), e145. (2021).

[6]. Carla Mouradian ,DialaNaboulsi, Sami Yangui, Roch H. Glitho, Monique J. Morrowand Paul A. Polakos. "A Comprehensive Survey on Fog Computing: State-Of-The-Art and Research Challenges". 1553-877X, 2017 IEEE.

[7]. Frank Alexander Kraemer, Anders EivindBraten, Nattachart Tamkittikhun and David Palma. "Fog Computing In Healthcare Review and Discussion". 2169-3536, 2017 IEEE

[8]. GeethaKurikala, K Gurnadha Gupta, A. Swapna. "Fog Computing: Implementation of Security and Privacy to Comprehensive Approach for Avoiding Knowledge Thieving Attack Exploitation Decoy Technology." 2017 IJSRCSEIT | Volume 2 | Issue 4 | ISSN: 2456-3307

[9]. GeethaKurikala, K GurnadhaGupta,A.Swapna "Fog Computing : Implementation Of Security And Privacy To Comprehensive Approach For Avoiding Knowledge Thieving Attack Exploitation Decoy Technology" IEEE(2017)

[10]. HadealAbdulaziz Al Hamid, SkMdMizanurRahman, M. ShamimHossain, Ahmad Almogren, and AtifAlamri,"A Security Model For Preserving The Privacy Of Medical Big Data In A Healthcare Cloud Using A Fog Computing Facility With Pairing-Based Cryptography, IEEE" [2017]

[11]. Karim Abouelmehdi, Abderrahim Beni, Hessane and Hayat Khaloufi. "Big Healthcare Data: Preserving Security and Privacy." Abouelmehdi et al. J Big Data (2018).

[12]. Malek Ben Salem and Salvatore J. Stolfo. "Decoy Document Deployment for Effective Masquerade Attack Detection." July 2011.

[13]. Mukherjee, M., Ferrag, M. A., Maglaras, L., Derhab, A., &Aazam, M. Security and privacy issues and solutions for fog. Fog and fogonomics: challenges and practices of fog computing, communication, networking, strategy, and economics, 353-374. (2020).

[14]. Rangel, T. C. Decoy technology in fog computing. Gsj, 7(4). (2019).

[15]. RongHeng Lin, ZeZhou Ye, HaoWangy, Budan "Chronic Diseases And Health Monitoring Big Data:ASurvey,IEEE"[2018]