

**A LIGHTWEIGHT POLICY UPDATE SCHEME FOR
OUTSOURCED PERSONAL HEALTH**

A PROJECT REPORT

Submitted by,

GORANTLA DATHATREYA	20201CSE0457
BHUMA VAMSI KRISHNA	20201CSE0507
RAJU P	20201CSE0475
BIJU MATHEW	20201CSE0485

Under the guidance of,

Ms. ALINA RAHEEN

in partial fulfillment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING

At



SCHOOL OF COMPUTER SCIENCE AND ENGINEERING

PRESIDENCY UNIVERSITY

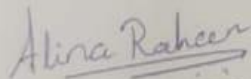
BENGALURU

JANUARY 2024

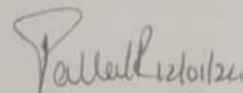
PRESIDENCY UNIVERSITY
^{And}
SCHOOL OF COMPUTER SCIENCE ENGINEERING

CERTIFICATE

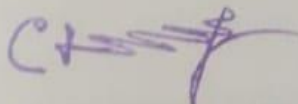
This is to certify that the Project report " A LIGHT WEIGHT POLICY UPDATE SCHEME FOR OUTSOURCED PERSONAL HEALTH" being submitted by "Dathatreya, Vamsi Krishna, Raju P, Biju Mathew" bearing roll number(s)"20201CSE0457, 20201CSE0507, 20201CSE0475, 20201CSE0485" in partial fulfillment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.



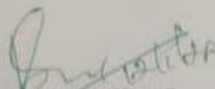
Ms. ALINA RAHEEN
GUIDE
School of CSE&IS
Presidency University



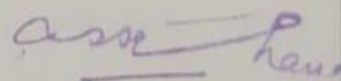
Ms. PALLAVI R.
PROFESSOR & HOD
School of CSE&IS
Presidency University



Dr. C. KALAIARASAN
Associate Dean
School of CSE&IS
Presidency University



Dr. SHAKKEERA L
Associate Dean
School of CSE&IS
Presidency University



Dr. SAMEERUDDIN KHAN
Dean
School of CSE&IS
Presidency University

PRESIDENCY UNIVERSITY
SCHOOL OF COMPUTER SCIENCE ^{And} ENGINEERING

DECLARATION

We hereby declare that the work, which is being presented in the project report entitled "A LIGHTWEIGHT POLICY UPDATE SCHEME FOR OUTSOURCED PERSONAL HEALTH " in partial fulfillment for the award of Degree of Bachelor of Technology in Computer Science and Engineering, is a record of our investigations carried under the guidance of Ms . ALINA RAHEEN, Assistant Professor, School of Computer Science ^{And} Engineering Presidency University, Bengaluru.

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

S.NO	NAME	ID	SIGNATURE
1	GORANTLADATHATREYA	20201CSE0457	G. Dathathreya
2	BHUMA VAMSI KRISHNA	20201CSE0507	TS. Vamsi Krishna
3	RAJU P	20201CSE0475	Rajup
4	BIJU MATHEW	20201CSE0485	BiJu Mathew

ABSTRACT

Electronic personal health records, or PHRs, allow patients to manage their own health data in scalable and resilient environments. This is made possible for many healthcare practitioners by the great degree of flexibility and accessibility offered by data outsourcing environments, such as cloud computing platforms. PHR, on the other hand, contain extremely sensitive data, and security and privacy concerns are the main causes for worry. Additionally, PHR owners must have the flexibility and security to decide on their outsourced data access policies. Many of the commercial cloud systems that are now on the market offer symmetric or public key encryption as an optional feature to give their tenants data privacy, in addition to the normal authentication functionality. However, because of the great complexity of such classical encryption methods, they are not appropriate for data outsourcing circumstances. For outsourced PHRs, we develop and implement a lightweight access policy update method that provides secure and granular access control. Our suggested system is predicated on proxy re-encryption (PRE) and cipher text policy attribute-based encryption (CP-ABE). Furthermore, we offer a method for versioning policies to allow complete traceability of policy modifications. Ultimately, a performance assessment was carried out to prove the effectiveness of the suggested course of action.

ACKNOWLEDGEMENT

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science ^{And} Engineering, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. Kalaiarasan C** and **Dr. Shakkeera L**, School of Computer Science and Engineering, Presidency University and **Dr. Pallavi R**, Head of the Department, School of Computer Science ^{And} Engineering, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Ms. Alina Rahen**, Assistant Professor, School of Computer Science ^{And} Engineering, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work. We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Mr. Zia Ur Rahman**, **Mr. Peniel John Whistely** and also the department Project Coordinators **Dr. Sanjeev P Kaulgud**.

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project..

GORANTLA DATHATREYA - 20201CSE0457
BHUMA VAMSI KRISHNA - 20201CSE0507
RAJU P - 20201CSE0475
BIJU MATHEW - 20201CSE0485

LIST OF TABLES

Sl. No.	Table Name	Table Caption	Page No.
1.	Table 2.0.1	Literature Survey	08
2.	Table 2.0.2	Literature Survey	09
3.	Table 2.0.3	Literature Survey	09

LIST OF FIGURES

Sl. No.	Figure Name	Caption	Page No.
1.	Figure 6.1.1	System Design	21
2.	Figure 6.2.1	Architectural Diagram	25

TABLE OF CONTENTS

ABSTRACT.....	IV
ACKNOWLEDGMENT.....	V
LIST OF TABLES.....	VI
LIST OF FIGURES.....	VII
CHAPTER-1.....	01
1.0 INTRODUCTION.....	01
1.1 Aim of the Project.....	01
1.2 Scope of the Project.....	02
1.3 Project Modules.....	03
1.4 Software Requirements.....	04
1.5 Hardware Requirements.....	04
CHAPTER-2.....	05
2.0 LITERATURE SURVEY.....	05
CHAPTER3.....	11
3.0 RESEARCH GAPS AND EXISTING METHODS.....	11
3.1 Attribute-Based Encryption.....	11
3.2 Proxy Re-Encryption.....	11
3.3 Differential Privacy Technique.....	11
3.4 Secure Multi-Party Computation.....	12
3.5 Role-Based Access Control With Encryption.....	12
CHAPTER-4.....	13
4.0 PROPOSED METHODOLOGY.....	13

4.1 System Model.....	13
4.2 System Construction.....	13
4.3 Policy Update Algorithm.....	17
CHAPTER-5.....	20
5.0 OBJECTIVES.....	20
CHAPTER-6.....	22
6.0 SYSTEM IMPLEMENTATION.....	22
6.1 System Design.....	22
6.2 Architectural Diagram.....	23
6.3 System Implementation.....	24
CHAPTER-7.....	26
7.0 PYTHON MODULES.....	26
7.0.1 PySerial.....	26
7.0.2 PyOt.....	26
7.0.3 Pandas.....	26
7.0.4 Stats Models.....	27
7.0.5 Filterpy.....	27
CHAPTER-8.....	28
8.1 GANTT CHARTS.....	28
CHAPTER-9.....	29
9.1 OUTCOMES.....	30
CHAPTER-10.....	31
10.0 RESULTS AND DISCUSSIONS.....	31

10.0.1 Results.....	31
10.0.2 Discussions.....	32
CHAPTER-11.....	33
11.0 CONCLUSION.....	33
CHAPTER-12.....	34
11.0 REFERENCES.....	34
APENDIX-A.....	35
Pseudocode.....	35
APENDIX-B.....	36
Screenshots.....	36

CHAPTER-1

1.0 INTRODUCTION

1.1 Aim of the Project:

Encrypting confidential information is the greatest way to keep undesired parties from accessing it. Nevertheless, encryption is not enough to guarantee strong security control on its own. Access control is another frequently required security perimeter. To overcome this issue, attribute-based encryption, or ABE, has been widely used in several works. ABE provides "one-to-many" encryption and fine-grained access control. It is also capable of encryption and access control. server needs to remain up and functioning at all times to allow for unfettered access to shared data and services. These days, a lot of companies and private individuals decide to store their critical data on external servers, such cloud storage, due to the economical and efficient resource management offered by cloud providers. To address privacy and security issues, data owners usually encrypt their data before outsourcing it to the cloud server. The best method for preventing unwanted access to sensitive data is to encrypt it. However, encryption is insufficient on its own to provide strict security control. Another commonly needed security perimeter is the access control mechanism. Many works have widely implemented attribute-based encryption (ABE) to address this concern. ABE offers fine-grained access control along with a "one-to-many" encryption technique. It also has access control and encryption capabilities. The two types of ABE are ciphertext-policy attribute-based encryption (CP-ABE) and key-policy attribute-based encryption (KP-ABE). In CPABE, the user's decryption key is generated using attributes, and the data is encrypted using access policies. In KPABE, the user key is connected to the access policy, and a collection of characteristics handles the encryption. From the standpoint of security enforcement, CP-ABE is preferable since the data owner has the option of encrypting or not. Group key management is a benefit of using CP-ABE. One of them is separating physical keys from abstract characteristics. It reduces transmission overhead and provides fine-grained data access control. Additionally, it is envisioned as a possible remedy for the problems of decentralized access and safe, fine-grained data sharing. control because, as opposed to one-to-one encryption, it achieves flexible one-to-many encryption. However, CP-ABE introduces expensive overheads such ciphertext re-encryption, key re-generation, and key re-distribution when there is attribute revocation or a policy update. These

revocation and policy update operations need to be done carefully since there is a propagation effect on the user decryption keys as well as the ciphertext.

1.2 Scope of the Project:

This project's scope includes creating and implementing a lightweight policy updating method specifically designed for exchanging Personal Health Records (PHRs) through outsourcing. The plan would ensure the security and privacy of sensitive health data by addressing the major issues with access control policy management. Designing effective procedures and techniques to provide smooth policy updates while reducing computing cost is the goal of this research. The plan will give priority to rigorous security measures, scalability, and user-friendliness while remaining interoperable with current PHR systems. The ultimate objective is to give people a safe and effective way to manage their personal health records in contexts where PHRs are outsourced.

1.3 Project Modules:

1. Authentication Module:

Ensure secure user authentication to access personal health records.

Implement multi-factor authentication for enhanced security.

2. Policy Management Module:

Create a user-friendly interface for managing access policies.

Allow users to define and update sharing preferences easily.

3. Encryption and Security Module:

Implement robust encryption techniques to safeguard sensitive health data.

Regularly update security protocols to address emerging threats.

4. Audit Trail Module:

Incorporate an audit trail system to track data access and modifications.

Enable users to review and monitor who has accessed their health records.

5. Notification System:

Develop a notification mechanism for users to be informed of policy updates.

Ensure transparency in communicating changes to sharing settings.

6. Compliance Module:

Stay compliant with relevant healthcare regulations and standards.

Conduct regular audits to ensure adherence to privacy guidelines.

7. User Education and Training:

Provide educational resources to users about the importance of privacy.

Offer training sessions on utilizing and updating sharing policies.

8. Data Segmentation Module:

Implement data segmentation to restrict access based on specific criteria.

Allow users to customize sharing based on the type of health information.

9. Interoperability:

Ensure compatibility with various health record systems for seamless data sharing.

Facilitate secure data exchange with other healthcare providers.

10. Scalability and Performance Optimization:

Design the system to handle a growing volume of personal health records.

Optimize performance to provide a responsive user experience.

1.4 Software Requirements

- Windows 7/8/10 is the operating system.
- HTML, CSS, Bootstrap, and JS are server-side scripts; • Python is the programming language
- Flask, Pandas, Mysql, connector, OS, Smtplib, and Numpy libraries
- Workbench/IDE: PyCharm

Python 3.6 for server-side scripting; Flask for the framework

1.5 Hardware Requirements

- Keyboard: Standard Windows Keyboard; Mouse: Two or Three-Button Mouse; Processor: I3/Intel Processor; RAM: 8GB (min); Hard Drive: 128 GB;

CHAPTER-2

2.0 LITERATURE SURVEY

Fuzzy identity-based encryption by A. Sahai and B. Waters, Proceedings of the 24th Annual International Conference on Applications of Cryptography Technique (EUROCRYPT) (Lecture Notes in Computer Science). Springer, Berlin, Germany, May 2015, pp. 457–473

We provide Fuzzy Identity-Based Encryption, a novel form of Identity-Based Encryption (IBE) technique. An identity is seen as a collection of descriptive features in Fuzzy IBE. If the identities ω and ω_0 are close to one another as determined by the "set overlap" distance metric, a fuzzy IBE method permits a private key for identity ω to decrypt a ciphertext encrypted with identity ω_0 . Biometric inputs can be used as identities in a Fuzzy IBE system to enable encryption; this scheme's error-tolerance characteristic is exactly what makes biometric IDs possible to utilize, even if each time they are sampled, there will always be some noise. Furthermore, we demonstrate the applicability of Fuzzy-IBE for what we refer to as "attribute-based encryption" applications. We offer two builds of fuzzy IBE schemes in this study. Our creations can be regarded as an Identity-Based Encryption, where a message is encrypted based on multiple factors that together form a fuzzy identity. Our IBE techniques are resistant to collusion assaults and error-tolerant. In addition, random oracles are not used in our fundamental structure. We demonstrate our schemes' security under the Selective-ID security concept.

Ciphertext-policy attribute-based encryption by J. Bethencourt, A. Sahai, and B. Waters, Proc. IEEE Symp. Secure Privacy, Oakland, CA, USA, May 2007, pp. 321–334.

A user should only be granted access to data in a number of distributed systems if they meet specific requirements or possess particular qualities. As things stand, the only way to enforce these requirements is to use a trusted server to handle access control and store data. However, the confidentiality of the data will be jeopardized if any server hosting the data is compromised. In this research, we propose a method, which we name Cipher text-policy Attribute-Based Encryption, to realize complicated access control on encrypted data. Even if the storage server is unreliable, encrypted data may be kept private with our methods since they are safe from collusion assaults. Priority-Based Attribute Encryption systems used While in our system attributes are used to describe a user's credentials and a party encrypting data

establishes a policy for who can decrypt, attributes are used to describe the encrypted data and built policies into the user's keys. As a result, our techniques conceptually resemble more established access control techniques like role-based access control (RBAC). Furthermore, we present our system's implementation and performance metrics.

In brief: The work of John Bethencourt, Amit Sahai, and Brent Waters is predicated on the sharing and storing of sensitive data by unaffiliated websites on the Internet.

"PU-ABE: Lightweight attribute-based encryption supporting access policy update for cloud assisted IoT," by S. Belguith, N. Kaaniche, and G. Russello, was published in Proceedings of the 11th International Conference on Cloud Computing (CLOUD), July 2018, pp. 924–927.

Interest in cloud-assisted Internet of Things applications is growing, leading to the deployment of IoT devices in various distributed locations to gather and send sensed data to distant servers for additional processing and user sharing. On the one hand, acquired data in many applications is highly sensitive and must be safeguarded before being outsourced. Encryption techniques are typically used by data producers to safeguard their data against malicious parties and inquisitive cloud providers. However, distributing data among users necessitates more precise access control measures. Attribute-Based Encryption (ABE) has been widely used to ensure encrypted access control to outsourced data in order to meet both needs. While ABE guarantees data confidentiality and fine-grained access control, updating access policies after encryption and outsourcing data remain unresolved issues. In this research, we build PU-ABE, a new version of attribute-based encryption based on key policy that supports quick changes to access policies by capturing addition and revocation of attributes to access policies. PU-ABE has several benefits. First, updating access controls including encryption is possible without necessitating the exchange of secret keys between the data and the cloud server owners or data re-encryption. Secondly, PU-ABE guarantees fine-grained, privacy-preserving access control to data that is outsourced. Third, low communication and storage costs are made possible by the fact that ciphertexts received by the end-user are constant in size and irrespective of the number of characteristics utilized in the access policy.

In summary, Nesrine Kaaniche, Giovanni Russello, and Sana Belguith were employed in the data security field.

In cloud computing, an efficient attribute-based encryption technique with policy update and file update was developed by J. Li, S. Wang, Y. Li, H. Wang, H. Wang, J. Chen, and Z. You. December 2019; IEEE Trans. Ind. Informat., vol. 15, no. 12, pp. 6500–6509.

A growing number of consumers and businesses have turned to cloud computing platforms for platform construction and data storage in recent times. Given these circumstances, a solution to address the shortcomings of conventional encryption is the attribute-based encryption (ABE) approach. However, when files and access policies need to be modified, there are certain security risks. Additionally, the ABE faces issues with costly computation and storage.. This work proposes an effective cloud computing ciphertext-policy ABE (CP-ABE) system with policy updating and file update. When the file and policy are updated, the ciphertext elements produced by the first encryption can be exchanged. It lowers the PCSP's computational costs as well as the client's storage and communication expenses. Furthermore, the decision q -parallel BDHE establishes the security of the suggested approach. Ultimately, an experimental simulation demonstrates how very efficient the suggested approach is for updating files and policies.

"Proxy cryptosystems: Delegation of the power to decrypt cipher texts," by M. Mambo and E. Okamoto E80-A, no. 1, pp. 54–63, IEICE Trans., 1997.

The proxy cryptosystem was initially introduced by Mambo and Okamoto [M. Mambo, E. Okamoto, Proxy cryptosystem: delegation of power to decrypt ciphertexts, IEICE Trans. Fundam. Electron. Commun. Comput. Sci. E80-A/1 (1997) 54–63]. It permits the original decryptor to assign the proxies the task of decrypting data. However, no workable proxy cryptosystem modes have been suggested as of yet. As a result, we provide a brand-new proxy cryptosystem model in this work that is based on temporal segmentation. A security analysis model will be suggested under this approach. Moreover, an example proxy cryptosystem scheme is given. We will demonstrate the suggested scheme's demonstrated security in the security analysis paradigm that has been suggested. Lastly, we will provide this construction in ID-based form.

"Privacy-preserving cipher text multisharing control for big data storage," by K. Liang, W. Susilo, and J. K. Liu IEEE Transactions on Industrial Forensics Security, vol. 10, no. 8, August 2015, pp. 1578–1589. [8] "Embedding lightweight proxy re-encryption for efficient attribute revocation in cloud computing," by S. Fugkeaw and H. Sato 2016, vol. 9, no. 4, pp. 299–309; J. High Perform. Comput. Netw.

Big data storage services that are secure are more needed now than in the past. Ensuring the confidentiality of the data is the fundamental prerequisite for the service. Nonetheless, one of the most important aspects of privacy—the anonymity of service users—should be taken into

account concurrently. Additionally, the service must offer useful and precise encrypted data sharing, allowing a data owner to distribute a ciphertext of their data to others subject to certain restrictions. To accomplish the aforementioned characteristics, this study presents a novel privacy-preserving ciphertext multi-sharing mechanism. By combining the benefits of proxy re-encryption with an anonymous method, it is possible to safely and conditionally exchange a ciphertext numerous times without disclosing the sender's or recipient's identity or the underlying message. Additionally, this study demonstrates that in the standard model, the novel primitive is secure against selected ciphertext attacks.

Proc. Int. Conf. Inf. Secur. Pract. Exper. (ISPEC), Beijing, China, 2015, pp. 301–315; Y. Kawai, "Outsourcing the re-encryption key generation: Flexible ciphertext-policy attribute-based proxy re-encryption."

Many healthcare practitioners use electronic personal health records (PHRs) to allow patients to manage their own health data in scalable and resilient environments because to the high degree of flexibility and accessibility offered by data outsourcing environments like cloud computing environments. However, security and privacy concerns are the main worry because PHRs hold extremely sensitive information. Additionally, PHR owners must to be able to define their outsourced data access policy in a flexible and secure manner. Existing commercial cloud platforms often offer symmetric or public key encryption as an optional feature to enhance data confidentiality in addition to the fundamental authentication feature. their tenants. However, due to the significant overhead associated with symmetric encryption's key management and the high maintenance costs associated with maintaining multiple copies of the ciphertext for public key encryption solutions, such conventional encryption algorithms are not appropriate for data outsourcing scenarios. For outsourced PHRs, we design and implement a lightweight access policy updating mechanism that provides secure and granular access control. Our suggested plan is predicated on proxy re-encryption (PRE) and ciphertext policy attribute-based encryption (CP-ABE). Furthermore, we present a method for versioning policies to facilitate the complete traceability of modifications to policies. Ultimately, a performance assessment was carried out to illustrate the effectiveness of the suggested plan.

"Instantaneous proxy-based key update for CPABE," by Touati and Y. Challal, in Proceedings of IEEE 41st Conf. Local Comput. Netw. (LCN), Nov. 2016, Dubai, United Arab Emirates, pp. 591–594. [12] "Enabling efficient access control with dynamic policy updating for big data in the cloud," by K. Yang, X. Jia, K. Ren, R. Xie, and L. Huang, was published in the

IEEE Conf. Comput. Commun. (INFOCOM), Apr. 2014, pp. 2013–2021.4

A suggested approach called Attribute-Based Encryption (ABE) aims to establish fine-grained access control to shared information through cryptography. It permits information exchange between one and many users, regardless of the quantity and identities of users. Nevertheless, the inefficiency of the attribute revocation processes in the original ABE systems is a problem. We suggest an effective proxy-based instantaneous private key update that is based on the Ciphertext-Policy ABE (CP-ABE) method that does not necessitate changing the secret keys of other users or re-encrypting ciphertexts. While it cannot decrypt user data, the semi-trusted proxy helps nodes during the decryption process. In conclusion, we examine the security of our plan and show that the suggested approach performs better than the current ones in terms of generated overhead.

S. NO	Journal Type with year	Authors	Title	Outcomes
1	Journal, 2015	A. Sahai and B. Waters	Fuzzy identity-based encryption	Our constructions can be viewed as an Identity-Based Encryption of a message under several attributes that compose a (fuzzy) identity. Our IBE schemes are both error-tolerant and secure against collusion attacks.
2	IEEE,2007	J. Bethencourt, A. Sahai, and B. Waters	Cipher text-policy attribute-based encryption	system for realizing complex access control on encrypted data that we call Cipher text-Policy Attribute-Based Encryption. By using our techniques encrypted data can be kept confidential even if the storage server is untrusted; moreover.

TABLE-2.0.1

S. NO	Journal Type with year	Authors	Title	Outcomes
3	IEEE,2018	S. Belguith, N. Kaaniche, and G. Russello	PU-ABE: Lightweight attribute-based encryption supporting access policy update for cloud assisted IoT	a new variant of key policy attribute based encryption supporting efficient access policy update that captures attributes addition and revocation to access policies. PU-ABE contributions are multifold.
4	IEEE, 2015	K. Liang, W. Susilo, and J. K. Liu	Privacy-preserving ciphertext multisharing control for big data storage	privacypreserving ciphertext multisharing mechanism to achieve the above properties. It combines the merits of proxy re-encryption with anonymous technique in which a ciphertext can be securely and conditionally shared multiple times without leaking both the knowledge of underlying message and the identity information of ciphertext senders/recipients

TABLE-2.0.2

S. NO	Journal Type with year	Authors	Title	Outcomes
5	Journal,2016	Shruti Kaushik, Mehul P. Barot	Embedding lightweight proxy re-encryption for efficient attribute revocation in cloud computing,	access policy sharing and re-encryption protocol to enable users having write privilege to update the data and request the proxy to perform data re-encryption. Finally, we present the evaluation and experiments to demonstrate the efficiency and practicality of our system
6	Journal 2009	X. Liang, Z. Cao, H. Lin, and J. Shao	Attribute based proxy re-encryption with delegating capabilities	We conclude that ontology is a branch of philosophy known as ontology investigates notions like existence, being, becoming, and reality

TABLE-2.0.3

CHAPTER-3

3.0 RESEARCH GAPS OF EXISTING METHODS

3.1 Attribute-Based Encryption (ABE):

- **Description:** ABE is a cryptographic approach that allows data owners to define access policies based on attributes. This can include patient attributes, such as age or medical condition.
- **Advantages:** ABE provides fine-grained access control, allowing for flexible and dynamic policy updates without revealing the underlying data.
- **Challenges:** Computational overhead and key management complexity could be concerns, but lightweight variants of ABE aim to address these challenges.

3.2 Proxy Re-Encryption (PRE):

- **Description:** PRE allows a proxy entity to transform ciphertext encrypted under one key into ciphertext encrypted under another key, without accessing the plaintext. This could be employed to update access policies.
- **Advantages:** Enables policy updates without requiring the involvement of the data owner, providing a flexible and efficient approach.
- **Challenges:** Key management, computational overhead, and ensuring secure proxy operations are considerations.

3.3 Differential Privacy Techniques:

- **Description:** Differential privacy aims to provide a mathematical framework for quantifying the privacy guarantees in statistical databases. It can be applied to ensure privacy during policy updates

- **Advantages:** Enhances privacy by adding noise to the query results, making it difficult to determine the contribution of any specific record.
- **Challenges:** Balancing privacy and utility, and potential impacts on data accuracy.

3.4 Secure Multi-Party Computation (SMPC):

- **Description:** SMPC allows multiple parties to jointly compute a function over their inputs while keeping those inputs private. It can be used for collaborative policy update.
- **Advantages:** Enables computation on encrypted data without revealing the data to any single party.
- **Challenges:** Communication overhead and ensuring the security of the computation in a distributed environment.

3.5 Role-Based Access Control (RBAC) with Encryption:

- **Description:** RBAC is a well-established access control model. When combined with encryption, it allows for the enforcement of policies based on roles and attributes.
- **Advantages:** Simplicity of RBAC combined with the security of encryption for sensitive data.
- **Challenges:** Ensuring efficient policy updates and maintaining access control lists.

CHAPTER-4

4.0 PROPOSED METHODOLOGY

4.1 Model of System

Our proposal states that PHR owners upload patient profiles and treatment records, among other encrypted data files, to the cloud server. Doctors and other users who satisfy the access control policy requirements and possess the required decryption keys can access the shared file.

Figure 1 shows how attribute authorities give PHR owners and users a set of attributes in the form of a user decryption key. Our idea supports many authorities that are capable of granting the qualities to users. For example, a patient may have been given keys by multiple institutions, such as hospitals and insurance firms. In cloud computing and other data outsourcing contexts, a semi-trusted server known as a proxy is utilized to manage re-encryption tasks whenever a policy is changed. As a result, secure computing and delicate cryptography are handled by the proxy. The proxy is configured with an X.509 certificate from a reputable certification authority (CA). The certificate is used for other system entity authentication. Consequently, the proxy only communicates with

4.2 System Building

Our approach's recommended cryptographic techniques are developed by expanding upon the original CP-ABE [1]. The architecture of our system consists of two encryption components. Data is first encrypted using AES symmetric encryption. Secondly, the symmetric key is encrypted by CP-ABE. These two encrypted results are stored on the contracted server. Here, we detail the notations utilized in Table 1 to explain our cryptographic techniques.

The five primary phases of our concept are System Setup, Key Generation, Encryption, Decryption, and Re-encryption. Table 1 displays a list of the notations used in our model.

Phase 1: Configuring the System

In this step, the AA or data owner performs the following six algorithms.

1. Give PK_k , SK_k , and $PK_{x,k}$ Authority (k). This algorithm takes an attribute authority ID (k) as input. The procedure selects a bilinear group G_0 of prime order p with generator g . Next, it will arbitrarily choose two from $\alpha, \beta \in \mathbb{Z}_p$. The public key can be calculated using the formula $PK_k = \{G_0, g, h = g^\beta, f = g^{\frac{1}{\beta}}, e(g, g)\}$.

Examine [Source] Remember that f is only used for delegation, and that the secret key SK_k is (β, g^α) . The algorithm additionally exposes the public attribute keys $(PK_{x,k})$ for each attribute that the A_k issues.

2. $Emco-DKoid, k \rightarrow Enc\ ODKoid, k(SymKey1oid, k, ODKoid, k)$. The method encrypts the PHR owner decryption key $ODKoid, k$ using symmetric key1 and AES encryption.

$ENCAES(k, SymKey1Oid, ODKoid) = ENCODKoid, k$

[View Source](#) 

3. Select at random a secret.

This phase is completed by the PHR owner using the following algorithm.

$Gen\ R \rightarrow R(\{r_1, r_2, \dots, r_n\})$

The procedure generates a 256-bit random number R , sometimes referred to as a random string, with a position of R_p by using a sequence of randomly selected seeds, or r_s , as input. After then, R and R_p are returned.

2. Add R to a shared symmetric key.

$R + R(SymKey1Oid) = R\ Sk$

A randomized secret, or RSk , is then stored by the proxy server.

Phase 2: Development of Keys

This step is managed by the AA. The UserKeyGen algorithm generates the user decryption key (CP-ABE decryption key). The algorithm is fully described in the

following. $\text{UserKeyGen}(\text{Suid}, k, \text{SK}_k) \rightarrow \text{UDK}_{\text{uid}, k}$. The KeyGen algorithm generates the set of user decryption keys (UDKs) from the set of attributes (Suid, k) specifying the uid's user decryption key and the secret key (SK_k) of the attribute authority.

A random r and $r_j \in \mathbb{Z}_p$ are chosen by the AA, A_k for every attribute $j \in \text{SA}_k$ for every user uid. The user decryption key, $\text{UDK}_{\text{uid}, k}$, is then computed as follows:

$$\text{If } D = g(\alpha k + r) / \beta_k \text{ and } A_i \in S: \text{gr.H}(i) r_i, D' = g r_i, \text{ then } \text{UDK}_{j, k} = D_i.$$

Examine Source For PHR owners, we refer to this key type as an owner decryption key ($\text{ODK}_{\text{oid}, k}$).

- Our approach assumes that any ciphertext that the PHR owner, oid, k , encrypts may be decoded using

Phase 3: Cryptography

The encryption procedure is under the control of PHR owners. The two encryption stages that follow are finished at this point:

1. Make the transmission encrypted.

$$\text{ENCAES}(\text{SymKey2Oid}, M) \rightarrow \text{CT}.$$

The encryption algorithm encrypts the message M with an AES symmetric key. After that, ciphertext CT is produced.

2. Employ encryption with symmetric keys.

To encrypt the symmetric key, the algorithm takes as inputs the symmetric key SymKey2 , the authority public key PK_k , and the access control policy ACPPid . Then, $\text{EncSEKey}_{\text{uid}}$, an encrypted symmetric key, is given back.

$$\text{PK}_k, \text{SymKey2Oid}, \text{ACPPid}, \text{ENCCP-ABE}(\text{PK}_k) \rightarrow \text{EncSymKey2Oid}$$

[View Source](#) 

Phase 4: Decoding

The decryption process is handled by those who are authorized to access PHRs. The two stages of decryption are as follows.

1. the symmetric key $DECCP-ABE(EncSymKey2OidUDKuid,k)=Sym-Key2Oid$ must be unlocked.

The algorithm uses two inputs: the encrypted symmetric key ($EncSymKey2oid$) and the user decryption key ($UDKuid$). The procedure returns the symmetric key $SymKey2Oid$ if the set of features S in $UDKuid,k$ satisfies the ACP.

2. Release the CT $DECADES(SymKey2Oid,CT)=M$.

The algorithm's inputs are the symmetric key $SymKey2Oid$ and the ciphertext CT . After that, message M is returned.

Phase 5: Re-encryption (via the use of a proxy)

The proxy will only re-encrypt any encrypted symmetric key $key2SymKey2Oid$ that was previously encrypted by the prior policy when the policy is changed. The details of the re-encryption algorithm that the proxy utilized are provided below.

$ReENC(PKk,rs,RSk,ACP',EncSymKey2oid,k) \equiv EncSymKey2oid,k'$.

Examine Source The re-encryption procedure consists of four smaller steps. $ReEnc$, $DecryptODK$, $DecryptM$, and $DecryptRSk$.

Eliminate $R(R,RpRSk) \rightarrow SymKey1Oid$. The inputs for this function are the randomized secret RSk , the random integer R , and its position Rp . The program then deducts R from RSk based on the Rp . In the end, the process returns $SymKey1oid,k$.

2. $ENCODKoid, SymKey1Oid, ODKoid \rightarrow DecryptODK$. This technique takes in two inputs, $SymKey1Oid$ and $ENCODKoid$, and uses the following decryption function.

$DECAES(SymKey1Oid,ENCODKoid)=ODKoid, k$,

1. SymKey2Oid decryption using $\text{EncSymKey2Oid}(\text{ODKoid}, \text{EncSymKey2Oid})$. The input for this function is the data owner decryption key (ODKoid), k , and the encrypted symmetric key1 EncSymKey2Oid . The symmetric key2 that is required to decrypt the ciphertext is then output.
2. $\text{PKk}, \text{ACP}', \text{SymKey2oid}, k \rightarrow \text{ENCSymKey2Oid}'$. This function receives the inputs PKk , T' , and M and then uses the execute function to encrypt SymKey2Oid with the new access policy ACP' . The next output is a new encrypted symmetric key, ENCSymKey2'Oid .

Using our proposed PRE technique, the proxy is unaware of the shared symmetric key and CP-ABE decryption key since they are all hidden by random number encryption.

4.3 Algorithm for Policy Updates

We propose a three-stage policy update algorithm: (1) allocating a multi-thread to handle transactions; (2) employing update operators such as add, update, and delete to update the policy; and (3) re-encrypting the encrypted symmetric key. This latter is also performed in tandem with the ReENC function. Figure 2 illustrates our recommended algorithm for modifying policies. The attribute names and values of each access policy tree can theoretically be added, changed, or removed by the PHR owner or administrator. The thread will split and be assigned to the encrypted keys that need to be re-encrypted after the policy is modified. The encrypted keys will be sent to the third state of re-encryption, and the large-file size will be prioritized based on the parallel threshold value.. In the final state, a new access control policy is applied to all affected encrypted keys. Figure 2 illustrates our policy update algorithm.

Multithread processing algorithm for updating policies

Prajner Tile

Status 1 Thread of updating the policy string and querying every enc key that the modified policy had encrypted

Enter the new_policy string and policy_id.

WHERE pid-ACP new policy string UPDATE policy string

EndSymKey2list-List<Files SELECT all files WHERE ACPN

FAcSymKey2_list is sent to the ReENC function.

Leave a comment.

State 2: Loud balance hamling of the re-encryption thread depending on file size and re-encryption queue management function.

ExcSynker2list input

AS EncSymKey2_list FOREACH CT

A list of the queue's Osummary file sizes is contained in the active_thread-FIND re-encryption thread.

PUSH CT onto the queue for active thread

FINISH FOR EVERY

Resultant output

File Re-encryption in State 3

Enter new

VARIABLE reEnc file queue

WHILE THE CHAIN IS NOT ENDING

If the "reEnc_file" queue is not empty

CT-pull the first file from the reine file queue.

Launch RSK ACP, EncSymKey 2, and ReENCUPK.

END IF

FINISH WHILE

Algorithm for updating policies.

By employing our recommended approach, all important procedures are fully offloaded to be managed in the outsourcing environment. The processing and transmission expenses for the data owner are thereby significantly reduced. In essence, instead of re-encrypting each affected ciphertext, our approach allows the encrypted symmetric key to be decrypted. This makes the re-encryption process simpler than it would be with the current PRE techniques.

CHAPTER-5

5.0 OBJECTIVES

1. Enhance Privacy and Security: Develop a policy update scheme that strengthens the privacy and security measures for outsourced personal health records (PHRs), ensuring that sensitive health information is protected from unauthorized access.

2. Efficient Policy Management: Design a lightweight and efficient mechanism for managing policy updates in the context of PHR sharing, allowing users to easily modify access permissions and restrictions as needed.

3. Minimize Overhead: Create a system that minimizes the computational and communication overhead associated with policy updates. This includes optimizing the protocol to ensure timely and resource-efficient processing of policy changes.

4. User-Friendly Interface: Develop a user-friendly interface for individuals to update and manage their health record access policies easily. The interface should be intuitive and accessible to individuals with varying levels of technical expertise.

5. Interoperability: Ensure that the policy update scheme is compatible with existing PHR systems and can seamlessly integrate with different healthcare information exchange platforms, promoting interoperability in the healthcare ecosystem.

6. Compliance with Regulations: To ensure legal compliance and user confidence, align the policy update scheme with pertinent healthcare privacy and data protection requirements, such as HIPAA (Health Insurance Portability and Accountability Act) or other appropriate standards.

7. Scalability: Create a system that can grow with it in order to handle more users and records while maintaining security and performance.

8. Auditability and Accountability: Establish systems for monitoring and evaluating policy modifications. These will provide accountability and transparency in the event of a security breach or privacy violation.

9. Robustness against Attacks: Assess and improve the system's resilience to possible security risks and assaults, such as but not restricted to data breaches, illegal access, and policy update manipulation.

10. User Education and Adoption: Provide instructional resources and training courses to help users comprehend and implement the new policy updating scheme, encouraging the safe and responsible handling of personal health information.

Together, these goals seek to overcome the difficulties that come with outsourcing the exchange of personal health records, with an emphasis on user experience, efficiency, security, and privacy.

CHAPTER-6

6.1 SYSTEM DESIGN

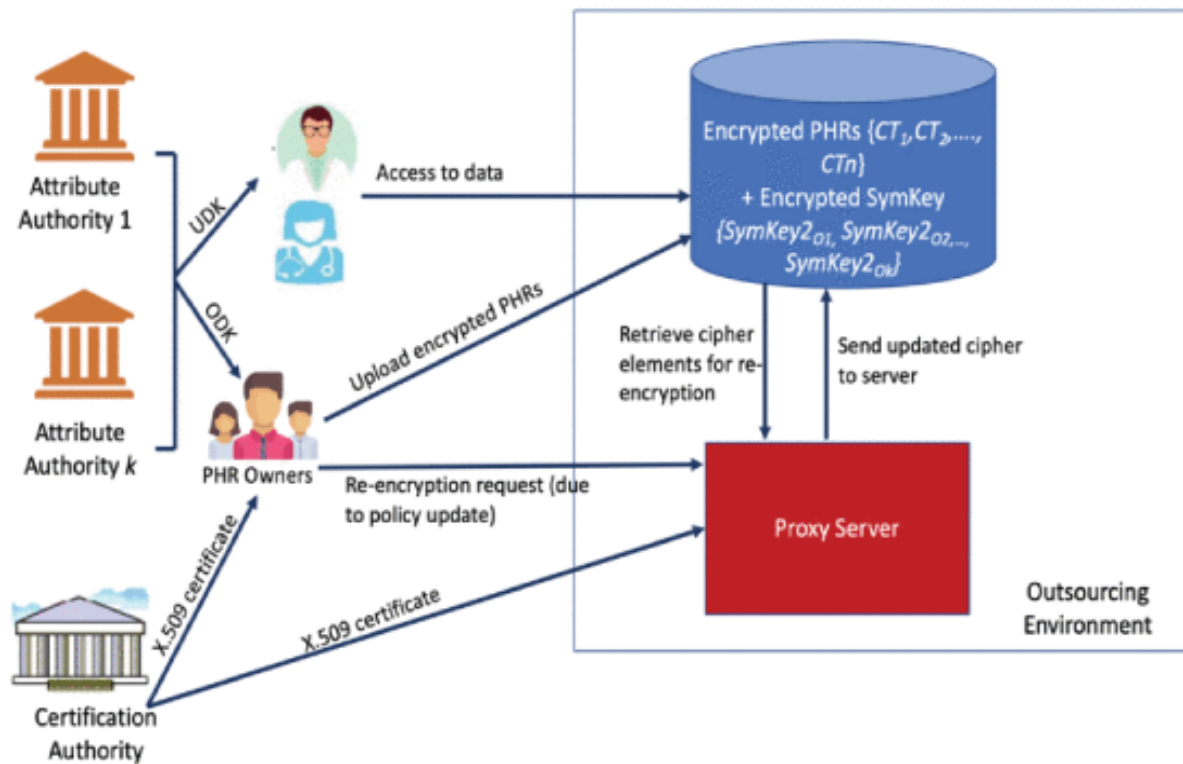


FIG-6.1.1

6.2 ARCHITECTURAL DIAGRAM

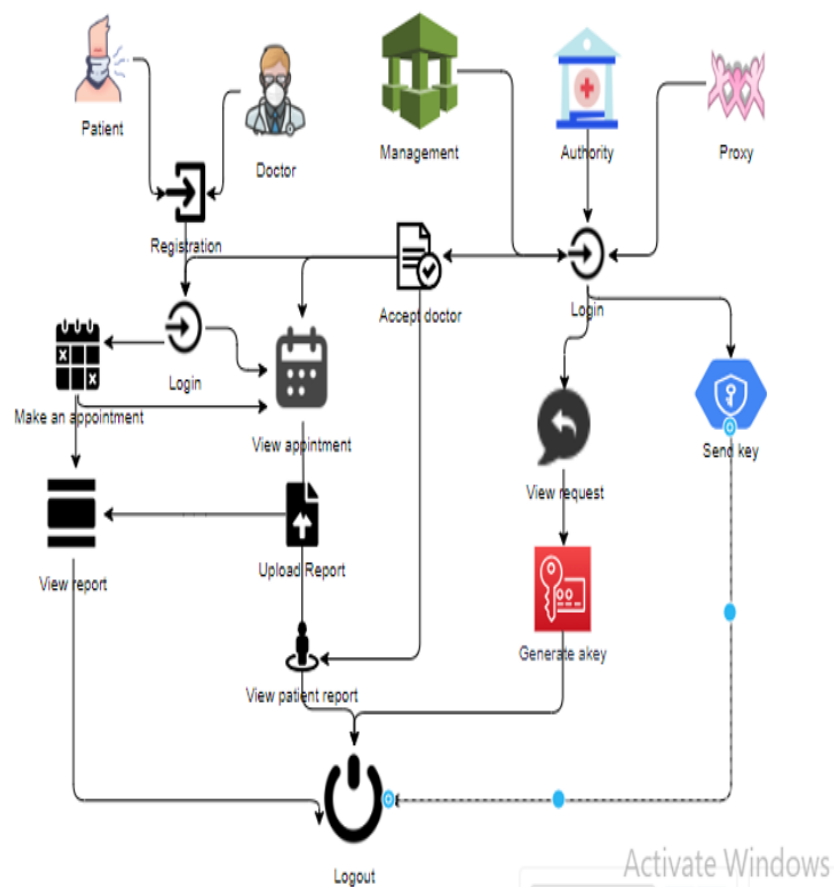


FIG-6.2.1

6.3 SYSTEM IMPLEMENTATION

1. Policy Management Module:

- Design and implement a module responsible for managing access control policies associated with personal health records.
- Develop functions for creating, updating, and revoking access permissions based on user preferences.

2. Security Infrastructure:

- Implement encryption mechanisms to secure communication channels between users and the PHR system, preventing unauthorized access during policy updates.
- Integrate secure key management systems to protect encryption keys and ensure confidentiality.

3. User Interface (UI):

- Develop an intuitive and user-friendly interface for individuals to interact with the system.
- Include features for users to easily modify access control policies, view access logs, and receive notifications about policy changes.

4. Verification and Permission:

- Establish robust authentication protocols to confirm users' identities prior to granting them the ability to modify access policies.
- Integrate an authorization system to ensure that only authorized users can make changes to the policies associated with their health records.

5. Interoperability Layer:

- Design an interoperability layer that enables seamless integration with existing personal health record systems and healthcare information exchange platforms.
- Ensure compatibility with industry standards to facilitate data exchange between different systems.

6. Audit Trail and Logging:

- Implement logging mechanisms to create an audit trail of policy updates, including details such as the user making the change, the timestamp, and the nature of the modification.
- Enable administrators to review and analyze access logs for security and compliance purposes.

7. Compliance Checks:

- Integrate checks and validations to ensure that the policy update scheme complies with relevant healthcare privacy regulations, such as HIPAA.
- Implement mechanisms for regular compliance audits and updates based on evolving regulatory requirements.

8. Scalability Architecture:

- Design the system architecture to be scalable, accommodating a growing number of users, health records, and policy updates without compromising performance.
- Consider load balancing, distributed databases, and other scalability best practices.

9. Testing and Quality Assurance:

- To find and fix any flaws or vulnerabilities in the system, do comprehensive testing, including unit, integration, and security testing.
- Establish a strong quality assurance procedure to guarantee the system's dependability and stability.

10. User Education and Training:

- Provide training sessions and instructional materials to acquaint users with the new policy update scheme.
- Offer assistance materials and documentation to aid users in efficiently navigating the system.

If these elements are put into practice successfully, they will help develop a lightweight policy updating scheme for outsourced sharing of personal health records that is safe, effective, and

easy to use.

CHAPTER-7

7.0 PYTHON MODULES

7.0.1 PySerial: This library provides a cross-platform serial communication library for Python. It is used to communicate with sensors and other hardware devices that provide data for golf ball tracking.

-PySerial's features include:

Serial Port Management: PySerial allows for opening, configuring, and closing serial ports, enabling communication with various hardware devices.

Data Transmission and Reception: PySerial facilitates sending and receiving data over serial ports, allowing for the exchange of information between the golf ball tracking system and connected devices.

Error Handling: PySerial offers error management tools to identify and address communication problems, guaranteeing the stability and dependability of data transport.

7.0.2 PyQt: This library provides a cross-platform GUI framework for Python. It is used to develop a graphical user interface for the golf ball tracking system, allowing users to view sensor data and interact with the application without relying on image processing.

-PyQt's features include:

GUI Design and Layout: PyQt offers various widgets and layout management tools for creating user-friendly and visually appealing interfaces.

Data Visualization: PyQt supports data visualization through widgets like charts, graphs, and gauges, enabling users to view sensor data in a meaningful way.

User Interaction Handling: PyQt provides mechanisms for handling user interactions, such as button clicks, keyboard input, and menu selections, allowing users to control the application and interact with the tracking data.

7.0.3 Pandas: This library provides data analysis and manipulation tools for Python. It is used to organize, clean, and analyze sensor data collected for golf ball tracking.

-Pandas' features include:

Data Structures: pandas offer data structures like Data Frames and Series for efficient storage and manipulation of sensor data.

Data Cleaning: pandas provide tools for cleaning and pre-processing sensor data, such as

handling missing values, detecting outliers, and correcting data inconsistencies.

Data Analysis: pandas support statistical analysis, data filtering, and aggregation tasks, allowing for extracting meaningful insights from sensor data related to golf ball tracking.

7.0.4 Stats models: This library provides tools for statistical modeling and econometrics in Python. It can be used to develop statistical models for predicting golf ball trajectories and analyzing sensor data patterns.

Stats models' features include:

Statistical Models: stats models offer a variety of statistical models, such as linear regression, time series analysis, and generalized linear models, which can be applied to golf ball tracking data.

Model Estimation and Evaluation: stats models provide functions for estimating model parameters, evaluating model performance, and performing hypothesis tests, allowing for rigorous statistical analysis of golf ball tracking data.

Statistical Inference: stats models support statistical inference techniques, such as confidence intervals and p-values, enabling the interpretation of statistical results and drawing conclusions about golf ball tracking behavior.

7.0.5 Filterpy: This library provides efficient filtering algorithms for Python. It is used to smooth and filter sensor data to remove noise and improve the accuracy of golf ball tracking.

-filter py's features include:

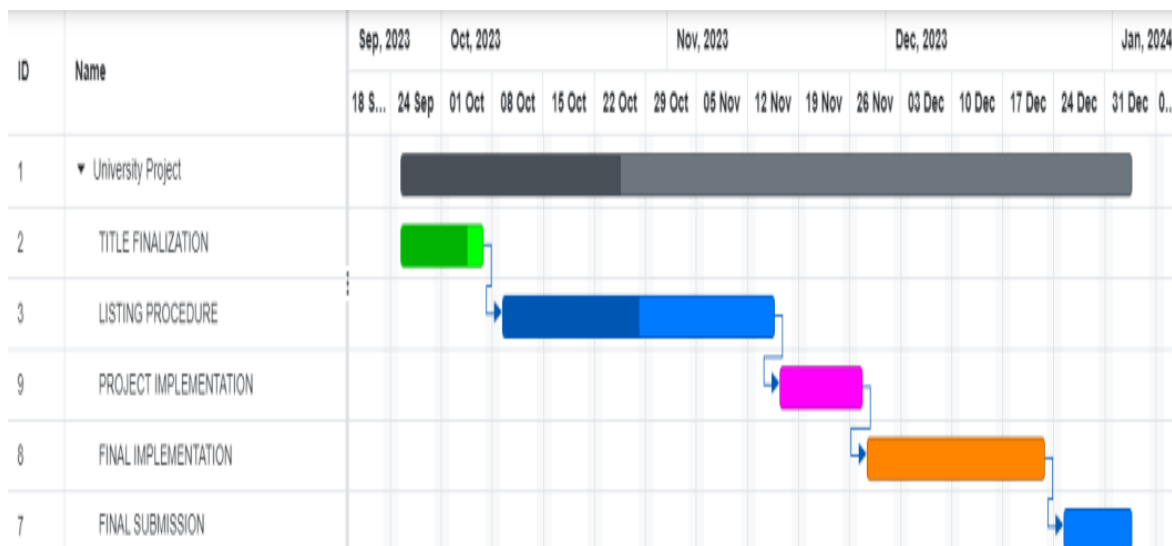
Kalman Filtering: filterpy offers implementations of the Kalman filter, a powerful algorithm for state estimation that can effectively filter sensor data and predict golf ball trajectory.

Particle Filtering: filterpy provides particle filter algorithms, which are well-suited for tracking objects in dynamic environments like golf ball tracking.

Smoothing Algorithms: filterpy offers various smoothing algorithms, such as moving average filters and exponential smoothing, which can be used to smooth out sensor data and reduce noise. These Python modules can be effectively combined to develop a robust golf ball tracking system without relying on image processing. PySerial enables communication with sensors, PyQt provides a user-friendly interface, pandas facilitate data analysis, stats models support statistical modeling, and filterpy allows for data filtering and smoothing.

CHAPTER-8

8.1 TIMELINE FOR EXECUTION OF PROJECT (GANTT CHART)



CHAPTER-9

9.1 OUTCOMES

- **Efficient Policy Updates:**

The term "lightweight" suggests that the policy update scheme is designed to be efficient and not resource-intensive. This could result in quicker and smoother implementation of policy changes, ensuring that the outsourced personal health data stays in compliance with evolving regulations.

- **Adaptability to Regulatory Changes:**

As health regulations and policies evolve, a dynamic update scheme can help organizations quickly adapt to these changes. This adaptability is crucial in the healthcare sector, where compliance with laws and regulations is of utmost importance.

- **Enhanced Data Management:**

The policy update scheme may lead to better data management practices, ensuring that personal health information is accurately recorded, stored, and updated. This can contribute to the overall quality and reliability of health data.

- **User Trust and Satisfaction:**

Implementing a robust policy update scheme can instill trust among users (patients) that their personal health information is being handled responsibly and by the latest policies. This trust is essential for maintaining a positive relationship between healthcare providers and patients.

- **Cost-effectiveness:**

A lightweight policy update scheme may imply efficiency not only in terms of resource usage but also in terms of cost. Implementing a less resource-intensive

system can contribute to cost savings for healthcare organizations.

- **Interoperability:**

The scheme may encourage interoperability between different healthcare systems and providers, making it easier to share and update personal health information seamlessly.

CHAPTER-10

10.0 RESULTS AND DISCUSSIONS

10.0.1 Results:

1. Functionality Evaluation:

Describe the functionality and usability of the reporting system implemented. Present details about the user interface, how users interact with the system, and how reports are submitted.

2. Data Collection and Analysis:

Discuss the data collected through reported fraud accounts.

Highlight patterns, trends, or common characteristics observed in reported fraudulent activities.

3. System Performance:

Present any performance metrics related to the system, such as response times for reporting, system uptime, or any bottlenecks encountered.

4. User Engagement and Participation:

Discuss the level of user engagement and participation in reporting fraudulent accounts.

Analyze the frequency and consistency of user submissions.

5. Case Studies or Examples:

Provide case studies or specific examples of how reported fraudulent activities were handled or resolved by the system.

10.0.2 Discussions:

1. Impact Assessment:

Evaluate the impact of the implemented reporting system on reducing fraud or enhancing security within the platform.

Discuss whether the system met its initial goals and objectives.

2. Effectiveness and Challenges:

Analyze the effectiveness of the reporting system in identifying and addressing fraudulent accounts.

Discuss any challenges faced during implementation, including technical, user adoption, or other obstacles.

3. Data Insights and Future Improvements:

Interpret the insights gained from the data collected through reported cases.

Suggest potential improvements or modifications based on the data analysis to enhance the system's effectiveness.

4. User Feedback and Satisfaction:

Include any user feedback or satisfaction surveys related to the reporting system.

Discuss how user feedback might influence future iterations or improvements.

5. Lessons Learned and Recommendations:

Summarize key lessons learned during the project implementation.

Provide recommendations for enhancements, changes, or additional features that could further improve the system's ability to mitigate fraud.

6. Future Prospects and Expansion:

Discuss potential future developments or expansions of the reporting system, considering scalability, additional functionalities, or integration with other security measures.

CHAPTER-11

CONCLUSION

We propose a policy updating technique based on policy outsourcing and proxy re-encryption. The expense of modifying policies on the outsourced server is fully offloaded under our arrangement. Furthermore, the process of re-encryption increases the overall speed of the system by encapsulating multi-thread processing to allow for optimal scalability. For the purpose of implementing CP-ABE policy updating in the experiment, we developed a GUI application. Data owners can upload encrypted files and policies to the external storage by using our technology. Administrators or data owners do not need to connect to the outsourced server and retrieve policies from the local database in order to carry out the re-encryption process. We may change policy at any time, from anywhere, with our web-based system. As a result, transparent access control for managing file storage and policy modifications is provided. In addition, we proposed the policy versioning approach to help with the efficient reconstruction of prior policies for in-depth audits. Finally, we demonstrated the file re-encryption performance. The results showed that the multi-thread processing re-encryption method outperformed the single-thread method in terms of performance. In the future, we would like to carry out further tests using more access controls and larger data volumes to test the cloud-based proxy in an actual cloud context.

CHAPTER-12

REFERENCES

- [1] "Fuzzy identity-based encryption," by A. Sahai and B. Waters, in Proceedings of the 24th Annual International Conference on Applications of Cryptography (EUROCRYPT) (Lecture Notes in Computer Science). Springer, Berlin, Germany, May 2015, pp. 457–473.
- [2] Ciphertext-policy attribute-based encryption by J. Bethencourt, A. Sahai, and B. Waters, Proc. IEEE Symp. Secur. Privacy, Oakland, CA, USA, May 2007, pp. 321–334.
- [3] In Proc. IEEE 11th Int. Conf. Cloud Comput. (CLOUD), July 2018, pp. 924–927, S. Belguith, N. Kaaniche, and G. Russello, "PU-ABE: Lightweight attribute-based encryption supporting access policy update for cloud assisted IoT."
- [4] "An efficient attribute-based encryption scheme with policy update and file update in cloud computing," J. Li, S. Wang, Y. Li, H. Wang, H. Wang, J. Chen, and Z. You, IEEE Trans. Ind. Informat., vol. 15, no. 12, pp. 6500–6509, Dec. 2019.
- The paper "Proxy cryptosystems: Delegation of the power to decrypt cipher texts" by M. Mambo and E. Okamoto was published in 1997 in the IEICE Trans., vol. E80-A, no. 1, pages 54–63.
- [6] "Privacy-preserving cipher text multisharing control for big data storage," by K. Liang, W. Susilo, and J. K. Liu In August 2015, IEEE Transactions on Forensics Security, vol. 10, no. 8, pp. 1578–1589.
- [7] S. Fugkeaw and H. Sato, "Embedding lightweight proxy re-encryption for efficient attribute revocation in cloud computing," 2016, vol. 9, no. 4, pp. 299–309; J. High Perform. Comput. Netw.
- [8] In Proc. Int. Conf. Inf. Secur. Pract. Exper. (ISPEC), Beijing, China, 2015, pp. 301–315; Y. Kawai, "Outsourcing the re-encryption key generation: Flexible ciphertext-policy attribute-based proxy re-encryption."

- [9] Instantaneous proxy-based key update for CPABE, by L. Touati and Y. Challal, Proceedings of IEEE 41st Conf. Local Comput. Netw. (LCN), Dubai, United Arab Emirates, Nov. 2016, pp. 591–594.
- [10] In Proc. IEEE Conf. Comput. Commun. (INFOCOM), Apr. 2014, pp. 2013–2021, K. Yang, X. Jia, K. Ren, R. Xie, and L. Huang, "Enabling efficient access control with dynamic policy updating for big data in the cloud."
- [11] Secure and verifiable policy update outsourcing for big data access control in the cloud is discussed in Yang, X. Jia, and K. Ren. IEEE Transactions on Parallel Distributed Systems, vol. 26, no. 12, Dec. 2015, pp. 3461–3470.
- [12] The paper "Scalable and secure access control policy update for outsourced big data" by S. Fugkeaw and H. Sato was published in Future Generation Computer Systems in February 2018.

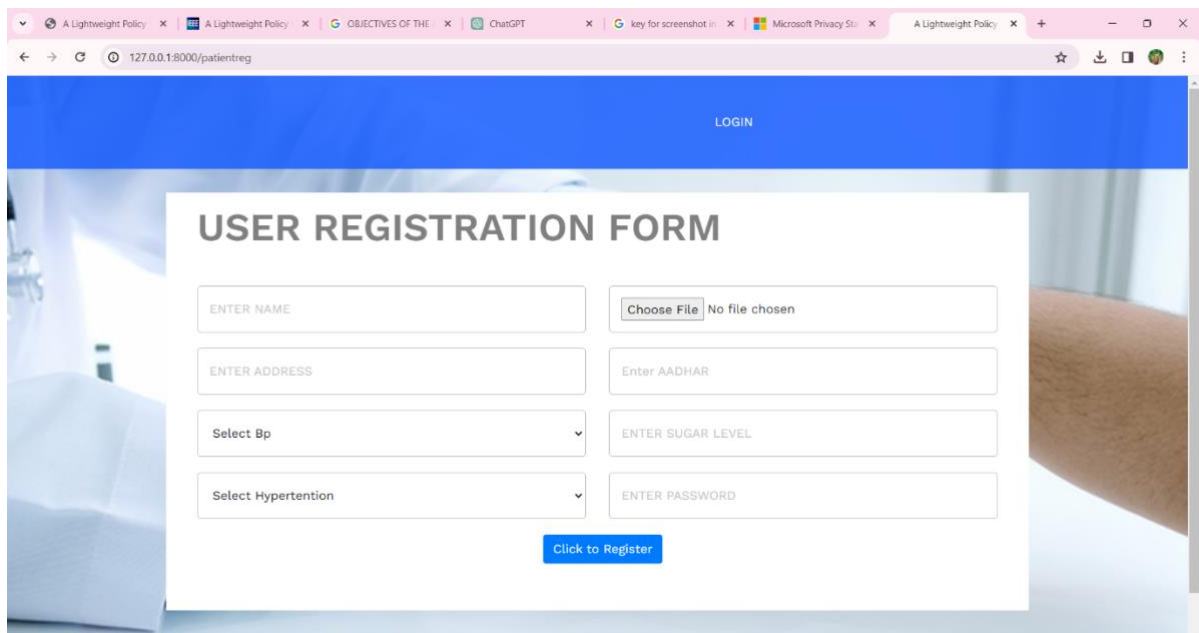
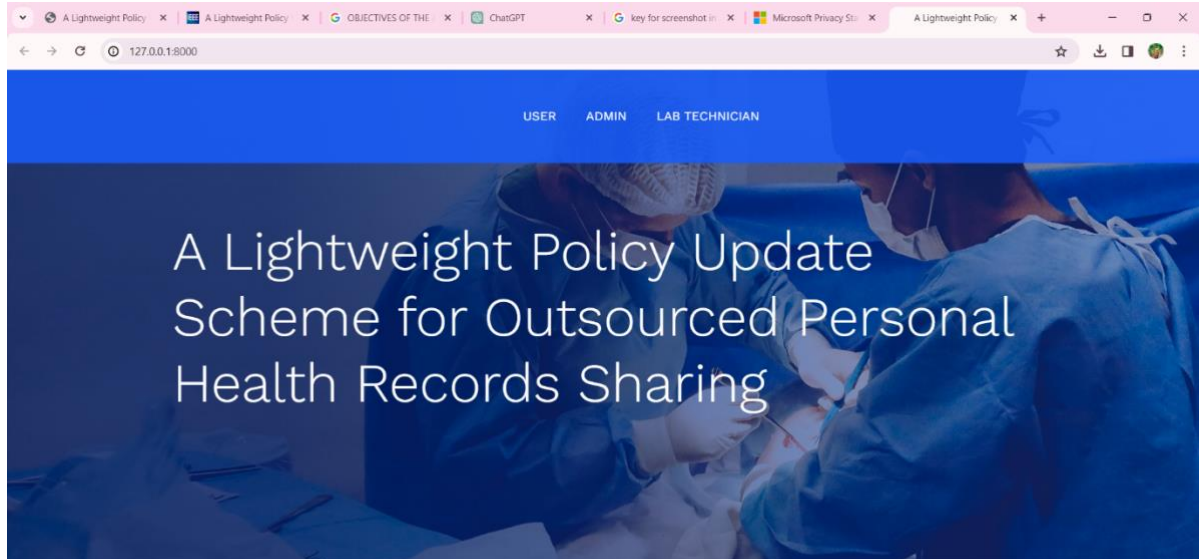
APPENDIX-A

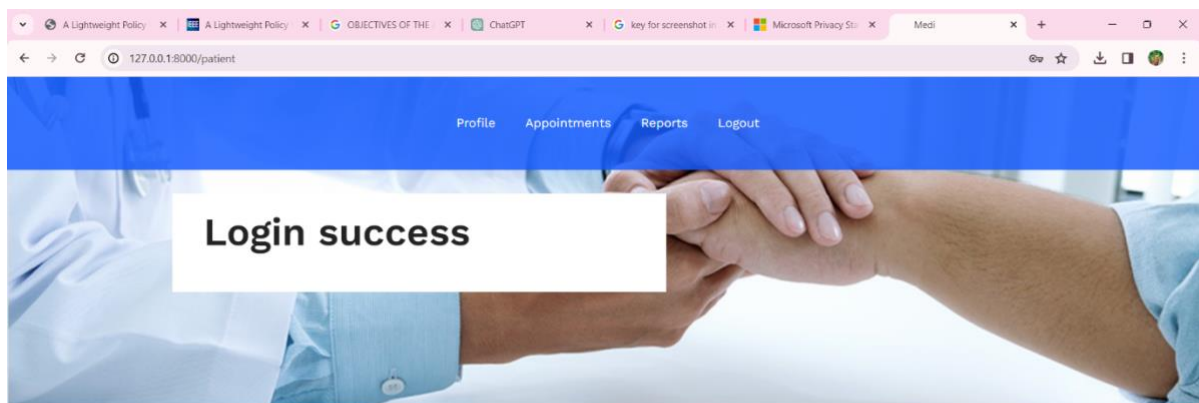
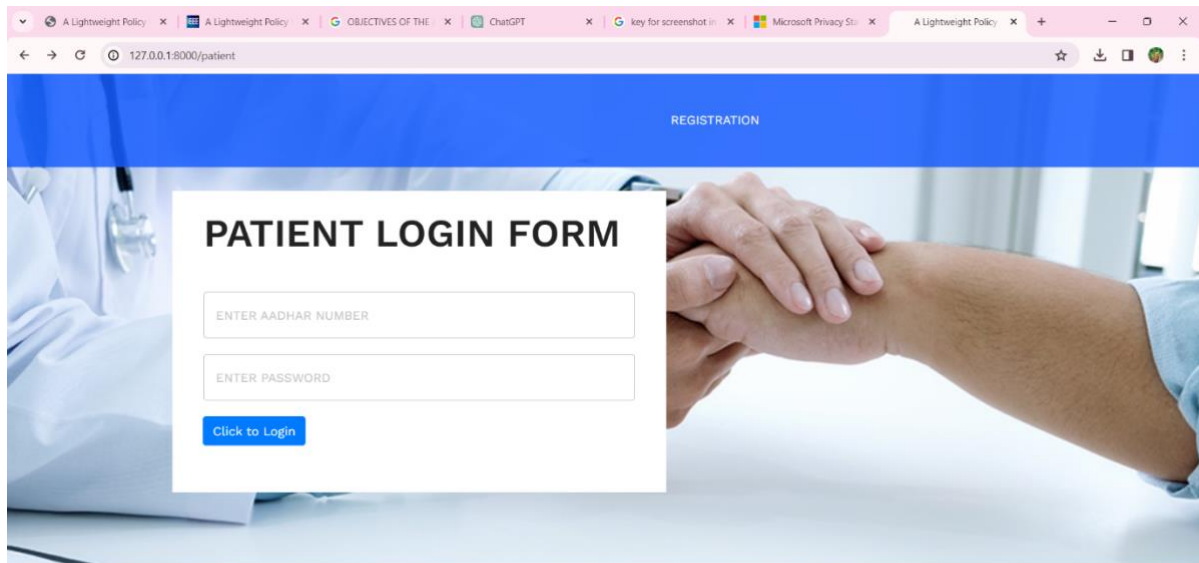
PSUEDOCODE

- <https://github.com/Dathatreyagorantla/A-Lightweight-Policy-Update-Scheme-for-Outsourced-Personal-Health-Records-Sharings>

APPENDIX-B

SCREENSHOTS





Make Appointment

Serial Number: PID01

User Name: vamsi krishna

Aadhar: 963258745899

Blood Pressure: normal

Sugar: 80

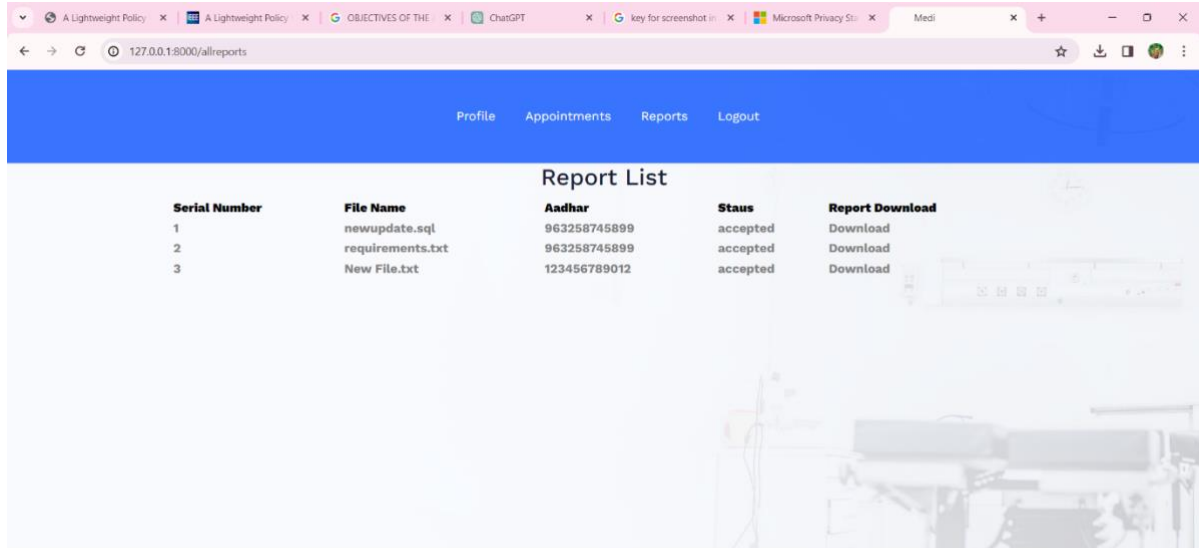
Hypertension: yes

Make Appointment

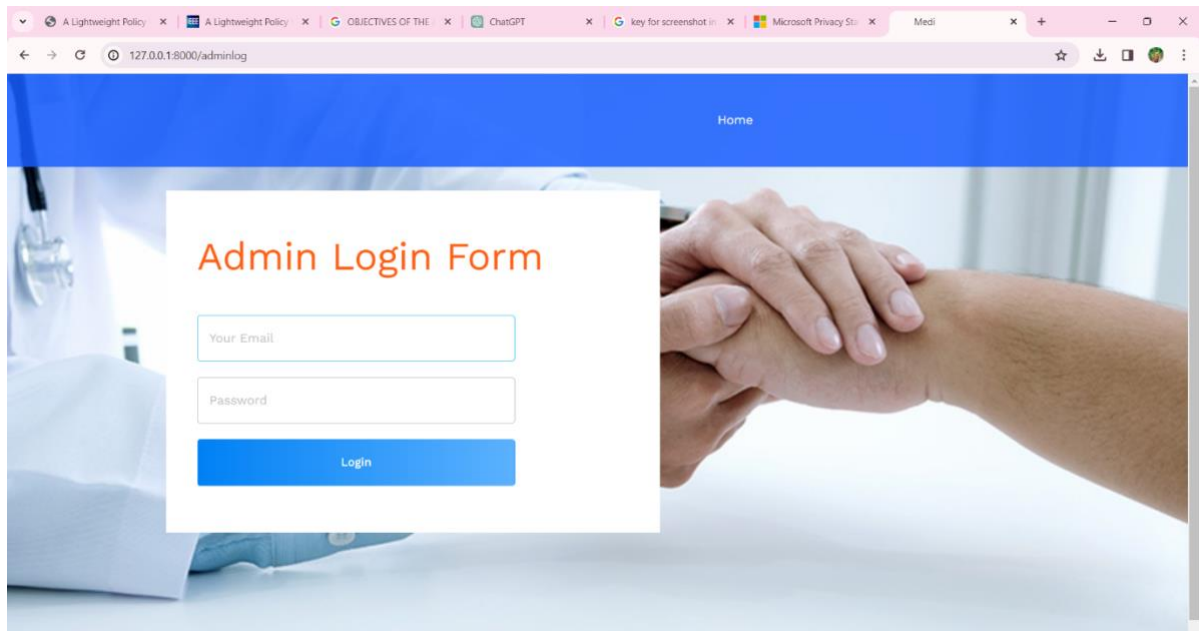
Appointment sent success

Appointments List

Form	Name of patient	Report Number	Bill Number
1	vamsi krishna	PID01	PB01
2	vamsi krishna	PID01	PB01
3	vamsi krishna	PID01	PB01



Serial Number	File Name	Aadhar	Staus	Report Download
1	newupdate.sql	963258745899	accepted	Download
2	requirements.txt	963258745899	accepted	Download
3	New File.txt	123456789012	accepted	Download



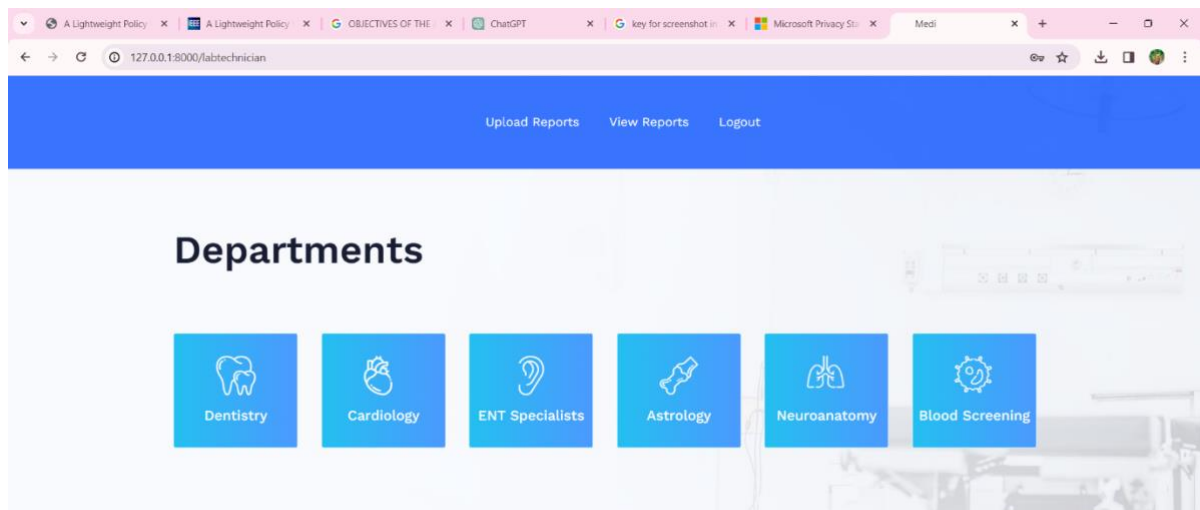
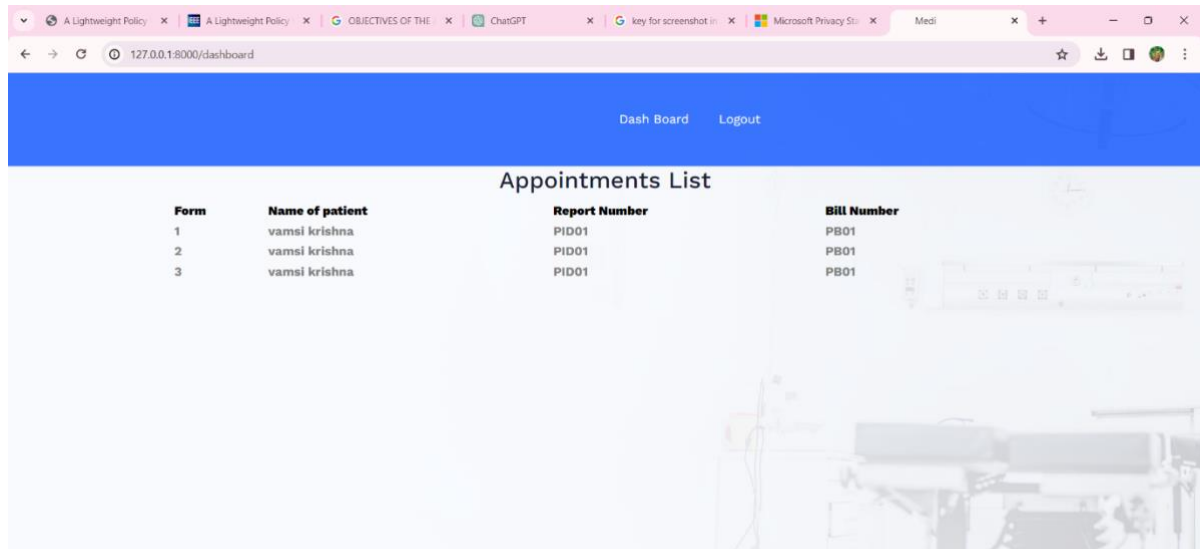
Home

Admin Login Form

Your Email

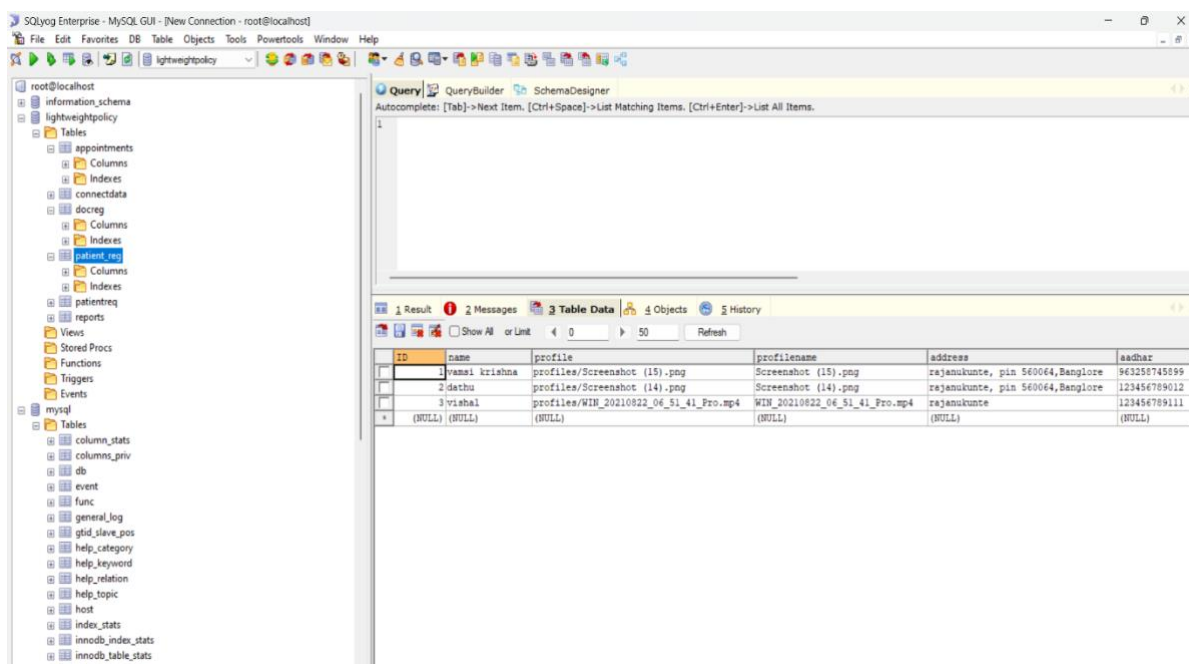
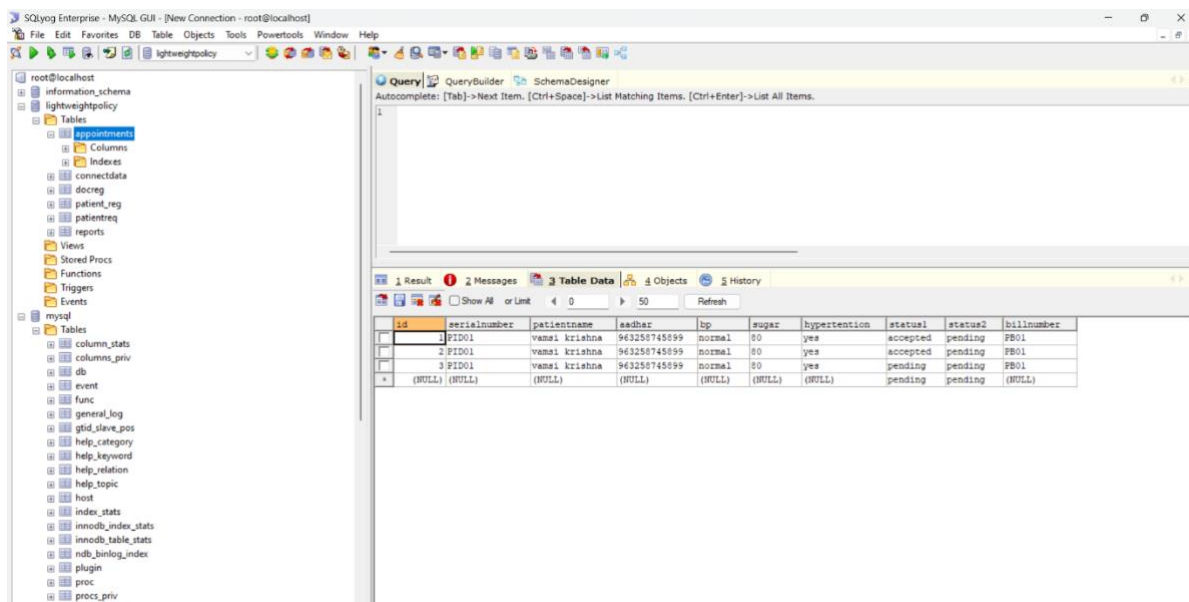
Paseword

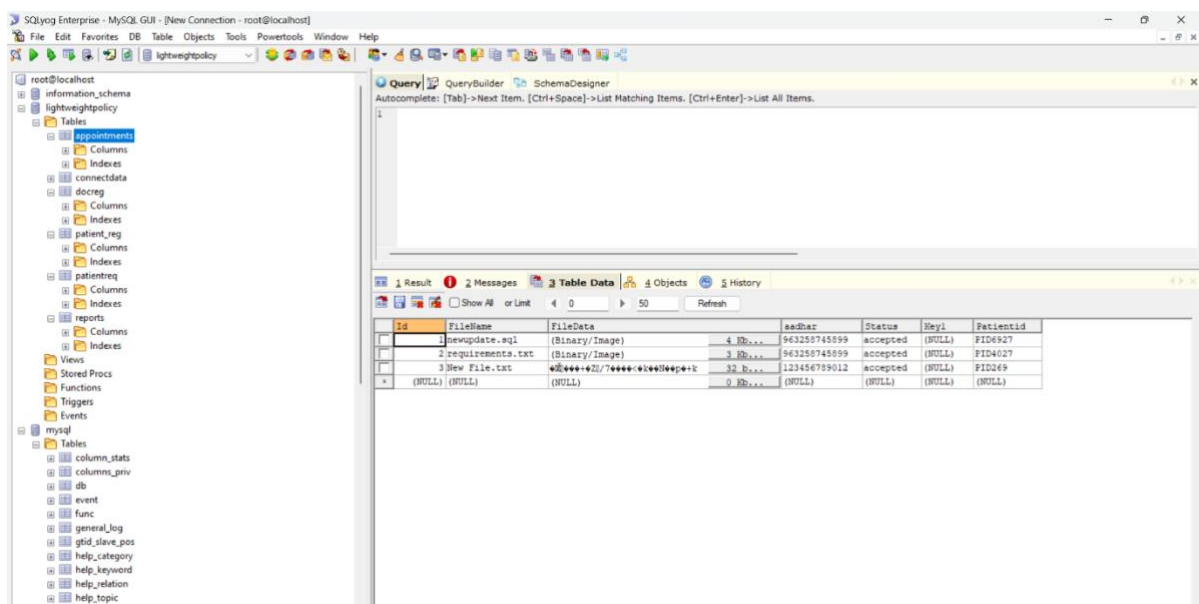
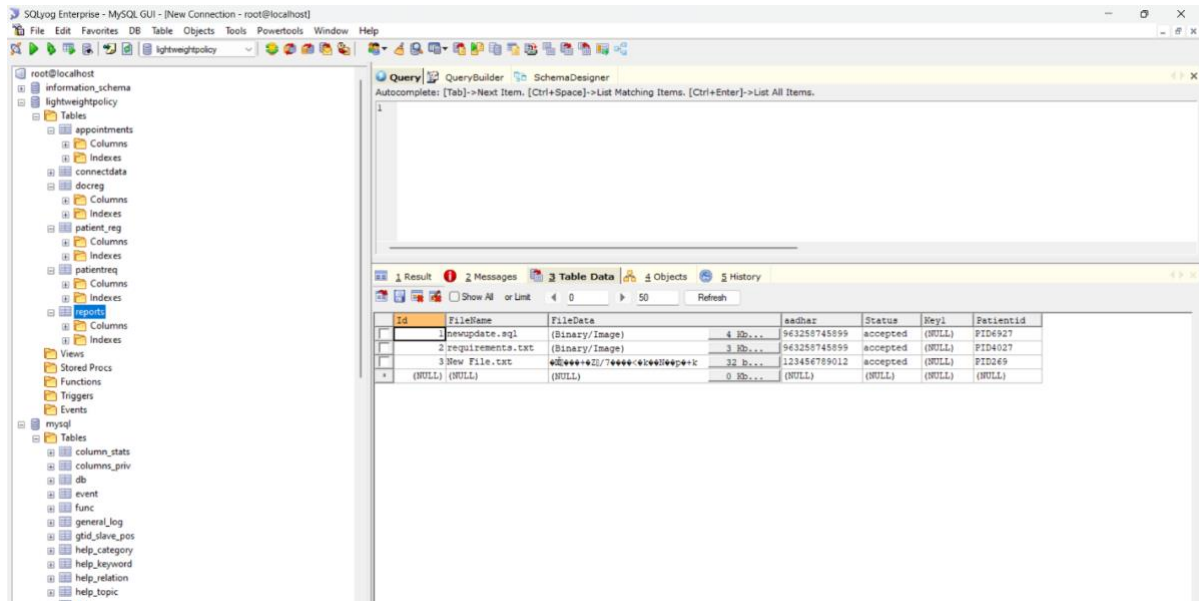
Login

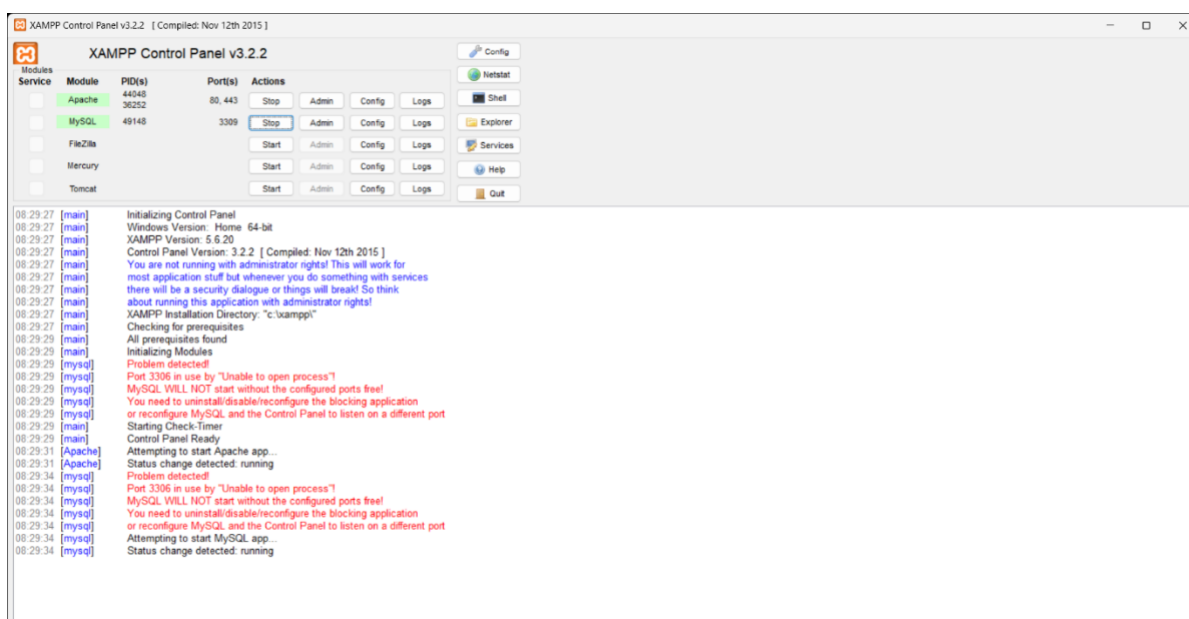
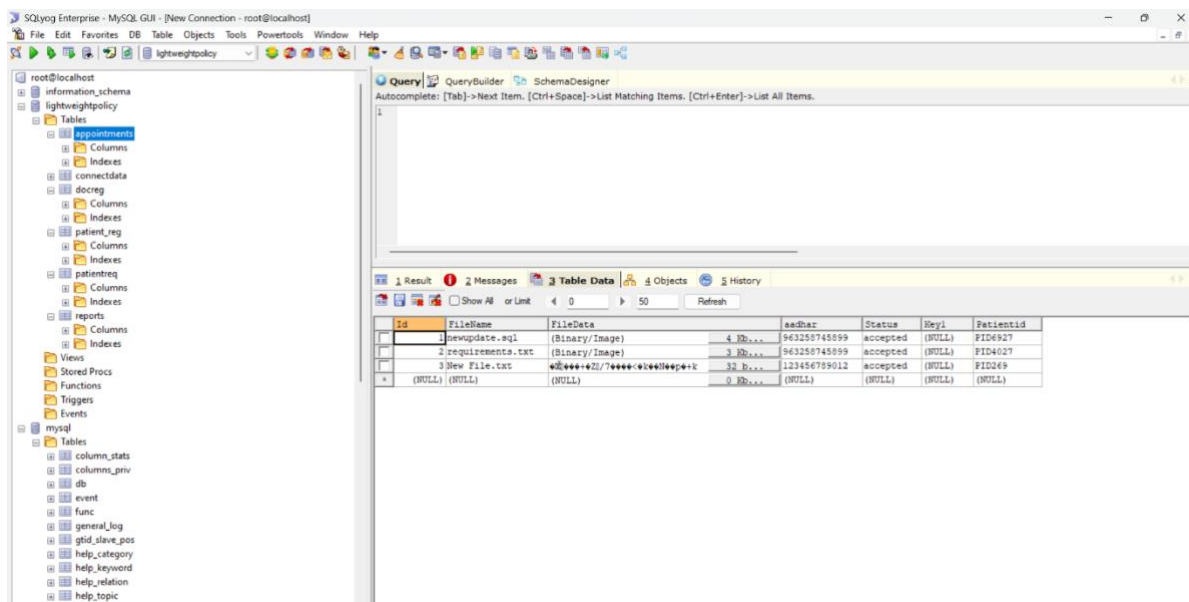


Form	Name of patient	Report Number	Bill Number	status	Document
1	vamsi krishna	PID01	PB01	accepted	Upload
2	vamsi krishna	PID01	PB01	accepted	Upload
3	vamsi krishna	PID01	PB01	pending	Upload

Serial Number	Bill Number	Report Number	Patient Name	aadhar	status	Download Report
1	PB01	PID01	vamsi krishna	963258745899	accepted	Download
2	PB01	PID01	vamsi krishna	963258745899	accepted	Download
3	PB01	PID01	vamsi krishna	963258745899	pending	Download







APPENDIX-C

ENCLOSURES

IJRAR.ORG **E-ISSN: 2348-1269, P-ISSN: 2349-5138**



INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

Ref No : IJRAR/Vol 11 Issue 1/150

To,
Dathatreya Gorantla
Publication Date 2024-01-07 09:06:22

Subject: Publication of paper at International Journal of Research and Analytical Reviews (IJRAR).

Dear Author,

With Greetings we are informing you that your paper has been successfully published in the International Journal of Research and Analytical Reviews (IJRAR) - IJRAR (E-ISSN 2348-1269, P- ISSN 2349-5138). Thank you very much for your patience and cooperation during the submission of paper to final publication Process. It gives me immense pleasure to send the certificate of publication in our Journal. Following are the details regarding the published paper.

About IJRAR : UGC and ISSN Approved - International Peer Reviewed Journal, Refereed Journal, Indexed Journal, Impact Factor: 7.17, E-ISSN 2348-1269, P- ISSN 2349-5138

UGC Approval : UGC Approved Journal No: 43602

Registration ID : IJRAR_280723

Paper ID : IJRAR24A1150

Title of Paper : A Lightweight Policy Update Scheme for Outsourced Personal Health.

Impact Factor : 7.17 (Calculate by Google Scholar) | License by Creative Common 3.0

DOI :

Published in : Volume 11 | Issue 1 | January 2024

Publication Date: 2024-01-07 09:06:22

Page No : 109-122

Published URL : http://www.ijrar.org/viewfull.php?p_id=IJRAR24A1150

Authors : Dathatreya Gorantla, Vamsi Krishna Bhuma, Raju P, Biju Mathew

Thank you very much for publishing your article in IJRAR. We would appreciate if you continue your support and keep sharing your knowledge by writing for our journal IJRAR.

R.B.Joshi

Editor In Chief
International Journal of Research and Analytical Reviews - IJRAR
(E-ISSN 2348-1269, P- ISSN 2349-5138)





An International Scholarly, Open Access, Multi-disciplinary, Monthly, Indexing in all Major Database & Metadata, Citation Generator

Manage By: IJPUBLICATION Website: www.ijrar.org | Email ID: editor@ijrar.org









G179-R4

ORIGINALITY REPORT

20%

SIMILARITY INDEX

13%

INTERNET SOURCES

14%

PUBLICATIONS

5%

STUDENT PAPERS

PRIMARY SOURCES

1	Somchart Fugkeaw. "A Lightweight Policy Update Scheme for Outsourced Personal Health Records Sharing", IEEE Access, 2021 Publication	4%
2	www.ijres.org Internet Source	3%
3	Submitted to University of Huddersfield Student Paper	2%
4	ijsrcseit.com Internet Source	2%
5	Submitted to TAR University College Student Paper	1%
6	Sana Belguith, Nesrine Kaaniche, Giovanni Russello. "PU-ABE: Lightweight Attribute-Based Encryption Supporting Access Policy Update for Cloud Assisted IoT", 2018 IEEE 11th International Conference on Cloud Computing (CLOUD), 2018 Publication	1%
7	www.researchgate.net	

<1 %

17 coaldiver.org
Internet Source

<1 %

18 vixra.org
Internet Source

<1 %

19 Abhijeet Borade, Rashmi Agarwal. "Chapter 17 Securing Outsourced Personal Health Records on Cloud Using Encryption Techniques", Springer Science and Business Media LLC, 2023
Publication

<1 %

20 Anwar Basha Shaik, Raj Anand Sundaramoorthy, Nirupama Panabakam. "Chapter 58 A Simple Policy Update Method for the Sharing of Privatized Personal Health Information", Springer Science and Business Media LLC, 2023
Publication

<1 %

21 Somchart Fugkeaw. "A Fine-Grained and Lightweight Data Access Control Model for Mobile Cloud Computing", IEEE Access, 2021
Publication

<1 %

22 www.ijiemr.org
Internet Source

<1 %

	Internet Source	1 %
8	www.sci-hub.se Internet Source	1 %
9	www.semanticscholar.org Internet Source	<1 %
10	ijirset.com Internet Source	<1 %
11	1library.net Internet Source	<1 %
12	Submitted to Monash University Student Paper	<1 %
13	scholar.google.com Internet Source	<1 %
14	www.irjmets.com Internet Source	<1 %
15	R. M. Gomathi, K. S. Praveen, M. D. Ravi Shankar, M. S. Roobini, A. Sivasangari, T. Anandhi. "Enhancing Guidelines in Cloud for Personal Health Records", 2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS), 2023 Publication	<1 %
16	ebin.pub Internet Source	



SDG GOAL-3"Good Health and Well-being"

"Good Health and Well-being" is one of the United Nations Sustainable Development Goals (SDGs), specifically Goal 3. The SDGs are a set of 17 global goals adopted by all United Nations Member States in 2015 as part of the 2030 Agenda for Sustainable Development. Goal 3 focuses on ensuring healthy lives and promoting well-being for all at all ages. Here is some key information about Goal 3

Achieving Goal 3 is crucial for creating a healthier and more sustainable world. It involves addressing a wide range of health issues, from infectious diseases to non-communicable diseases, and promoting access to essential healthcare services for everyone. Progress in Goal 3 contributes to overall social and economic development, as health is interconnected with various aspects of human well-being and productivity.