# BONAFIDE CERTIFICATE

Certified that this project report **“A SAFE AND RELIABLE IDENTITY BASED HEALTHCARE SYSTEM”** is the bonafide work of **“SOWNDARYA K(211420205151), THARANI.R(211420205169), YOGITAA DEVI R.C(211420205185)”** who carried out the project under mysupervision.

| | |
|---|---|
| **SIGNATURE** | **SIGNATURE** |
| **Dr. M. HELDA MERCY M.E., Ph.D.,** | **Mrs. MUTHULAKSHMI** |
| **M.Tech.,(Ph.D) HEAD OF THE DEPARTMENT** | **ASSOCIATE PROFESSOR** |
| **(SUPERVISOR)** | |
| Department of Information Technology | Department of Information |
| TechnologyPanimalar Engineering College | Panimalar Engineering College |
| Poonamallee, Chennai - 600 123 | Poonamallee, Chennai - 600 123 |

Submitted for the project and viva-voce examination held on _______________

| | |
|---|---|
| ________________ | ________________ |
| **INTERNAL EXAMINER** | **EXTERNAL EXAMINER** |

# DECLARATION

I hereby declare that the project report entitled “**A SAFE AND RELIABLE IDENTITY BASED HEALTHCARE SYSTEM”** which is being submitted in partial fulfillment of the requirement of the course leading to the award of the ‘Bachelor of Technology in Information Technology’ in **Panimalar Engineering College, An Autonomous institution Affiliated to Anna University- Chennai** is the result of the project carried out by me under the guidance and supervision of **Mrs. K.MUTHULAKSHMI, M.TECH.,(Ph.D) Associate Professor in the Department of Information Technology**. I further declared that I or any other person has not previously submitted this project report to any other institution/university for any other degree/ diploma orany other person.

(**Sowndarya.K**)

**(Tharani.R)**

**Date:**

**(Yogitaa Devi.R.C)**

**Place:** Chennai

It is certified that this project has been prepared and submitted under my guidance.

Date: (**Mrs. K.MUTHULAKSHMI**
**M.Tech.,(Ph.D)**

Place: Chennai (Associate Professo

# ACKNOWLEDGEMENT

A project of this magnitude and nature requires kind co-operation and support from many, for successful completion. We wish to express our sincere thanks to all those who were involved in the completion of this project.

Our sincere thanks to **Our Honorable Secretary and Correspondent, Dr. P. CHINNADURAI, M.A., Ph.D.,** for his sincere endeavor in educating us in his premier institution.

We would like to express our deep gratitude to **Our Dynamic Directors, Mrs. C. VIJAYA RAJESHWARI and Dr.C.SAKTHI KUMAR, M.E.,Ph.D** and **Dr.SARANYA SREE SAKTHIKUMAR., B.E., M.B.A.,Ph.D** for providing us with the necessary facilities for completion of this project.

We also express our appreciation and gratefulness to **Our Principal Dr. K. MANI, M.E., Ph.D.,** who helped us in the completion of the project. We wish to convey our thanks and gratitude to our head of the department, **Dr. M. HELDA MERCY, M.E., Ph.D.,** Department of Information Technology, for her support and by providing us ample time to complete our project.

We express our indebtedness and gratitude to our staff in charge, **Mrs.K. MUTHULAKSHMI M.Tech.,(Ph.D)** Associate Professor, Department of Information Technologyfor her guidance throughout the course of our project. Last, we thank our parents and friends forproviding their extensive moral support and encouragement during the course of the project.

# TABLE OF CONTENTS

# ABSTRACT

Nowadays, Cloud computing is a highly demanding field as various professionals, individuals and even various multinational companies are seamlessly utilizing public/private cloud for their day-to-day requirements. Most of the cloud solutions involve administrators at the backend, who are ensuring the privacy of data in the multi-tenant cloud infrastructure. Usually, a data owner stores data in the cloud which will get encrypted and decrypted in terms of their registered cloud standards. But if unregistered users want to share these stored data in the cloud with multiple users means it seems difficult to accomplish. To overcome these measures, TimeStamp based IBE system can be utilized. This approach helps in maintaining the secure shareability of data. In this proposed solution, data stored in the cloud gets identified and will be shared with of utilizing unique identities, which allows/denies access and eliminates the need to manage numerous certificates in a secure distributed environment.

The implementation of the proposed method deals with an identity-based, integrated multi-user system to store massive medical records in a secure cloud database and involves retrieving the medical records by Quick Response code which will be quick and widely acceptable. This solution can provide physicians with fast and more comprehensive access to patients‟ medical info in times of crisis and these approaches reduce administrative burden and aim at developing a cost-effective approach to assist doctors in an effective drug prescription, which maintains the integrity of data inside the secure communication channel. The experimental results show that the proposed method provides a significant way to accomplish a secure channel for medical records sharing with the defined algorithms.

# LIST OF TABLES

# LIST OF FIGURES

# LIST OF ABBREVIATIONS

| | |
|---|---|
| IBE | Identity Based Encryption |
| PKG | Private Key Generator |
| TS | Time Stamp |
| TC-IBE | Time Stamp Based Cocks Identity Based Encryption |
| PHI | Personal Health Information |
| PII | Personal Identifiable Information |
| RBAC | Role-Based Access Control |
| DS | Dynamic Salt |
| KF | Key Factor |
| HIPAA | Health Insurance Portability and Accountability Act |
| BF | Blowfish |
| CP | Cocks Parameter |
| PES | Proposed Encoding Scheme |

# CHAPTER 1

# INTRODUCTION

## 1.1 OVERVIEW OF THE PROJECT

Many organizations and individuals are engaged with Cloud computing due to its tremendous benefits of flexible storage services and high availability. These technologies added advantages to rural population as these allow sharing data among diverse locations and eliminates the need to manage and maintain data in local servers. These techniques ignite privacy issues and security issues of data. As the data owner hosts data in Cloud environment, they lose physical contact with data and they are concerned about their sensitive data hosted in Cloud platforms. These personal credentials shouldn"t be handed over to malicious users.

To avoid the attackers from stealing data, the data owners will employ cryptographic algorithms to encrypt data before they are outsourced to cloud environment. The cloud storage will contain data in cipher-text format and specific users with appropriate data decryption keys only able to decipher the data. These methodologies are widely acceptable but these standards will not meet proposed data sharing requirements as there are various cryptography algorithms to rely on and he/she can't share data with users rely with different cryptographic algorithm.

This arises forward/backward compatibility issues on long run and integrity of data is not taken over consideration on these mechanisms. So, to ensure confidentiality and integrity among data in various cloud platforms and to facilitate sharing of data among multiple users TC-IBE scheme will be advantageous. And these systems will be applicable to Healthcare sectors. As these sectors involve voluminous amount of data and third party entities associated with patients" health data, on establishing these secure sharing approach will benefit various hospital authorities and helps to share recorded health reports, diagnosis data among entities seamlessly.

The proposed TC-IBE scheme intends to exploit the IT technologies to establish secure sharing channel among hospital authorities and other third party entities to share personal health records of patients for further diagnosis or for extended care. This also helps to provide on-time emergency services to the people in the rural villages using secure communication channel helping to get advices from doctors across nation and to avail prescriptions based on the data collected from the past health records, thus supporting the welfare of the people.

## 1.2 NEED FOR THE PROJECT

Deploying an interactive multi-user sharing system with secure encryption algorithms will ensures the authenticity of the objects involved in the network. Identity based Approach allows the various entities to engage in the network to read/query the data.

Utilization of Private Key Generator encounters trust issues on the end-user sides. An efficient approach for establishing trust on employing a Hierarchical Also Trust model will ensure liabilities of PKG in Identity Based Encryption.

After expiration of access window will avoid replay attacks and safeguards the truthiness‟ of the information stored.The proposed system eliminates the demerits of managing voluminous amount of keys in public key infrastructure and set up of Private Key Generator with hierarchical trust model overcome the issues with of certifying authorities.

Sharing the message will encrypted with public key of receiver and private key of receiver can only able to decipher the encrypted message. healthcare environment, these services will benefit authorities of hospitals to share and receive data records for their communication and allow diagnosis of patients even in rural environments with utmost care.

Usually retrieving a patient‟s information through database can be maintained within the group of private hospitals and organizations which lack in integrity of data and not usually up to date in terms of maintained health record details. But patients will have their health records stored in diverse care centres where they are treated.These systems must agree upon key management prior to communication. Medical History of an individual is primary upon Diagnosis of his/her health discomfort.

## 1.3 OBJECTIVE OF THE PROJECT

To design an identity-based, integrated multi-user system to store massive medical records in a secure cloud database & retrieving the medical records by Quick Response code which will be quick and widely acceptable.

To provide physicians with fast and more comprehensive access to patients‟ medical info in times of crisis. This in turn reduce administrative burden.

To facilitate health data sharing in established system, various privileges

are configured on basis of their position in hierarchy and right to access records are maintained by role based access control system.

To develop a cost effective approach to assist doctors in an effective drug prescription and which maintains integrity of data.

To share access and rely on Private key generator with time bound access will accomplish secure communication channel for data sharing.

To engage with users upon authentication and permits the sharing with known identities of the recipient.

To encounter the environment and querying of data access can‟t be easily structured in these systems.

To share the encrypted data ( as of data privacy is concerned ) among a network of n users will complicate the Record Sharing platform.

To utilize in large scale industries for sharing data in wide range without the fear of tampering of data.

## 1.4 SCOPE OF THE PROJECT

In order to share encrypted data to vast amount of users, the encrypted data should be able to decrypt at faster rate but the management of public key for individual users will complicate the system as it involves every user to be pre organized with public and private key setup. But with the help of Identities, the centralized PKG will generate private key for users who enter the network of sharing with the help of master private key.Also the system will be dependent upon Hierarchical trust model involving Hospital authorities to ensure liabilities and confidentiality upon the concerned Health reports of the patients.

Getting past health recordsin another health care centres seems time consuming process. Quick assistance is needed in emergency cases and if the proposed system can provide that inevitable information in high time, then the treatment can be lot more easier and also facilitates safe and secure health record management and maintenance in further days also.

The quick retrieval of the patient‟s health records is challenging as of their treatment can be facilitated by various hospital management sectors across globe and various hospital sectors employ various approaches to store their patients‟ received care details related to their diagnosis or prescriptions.

Key agreement infrastructure plays fundamental role in establishing secure communication among diverse entities, but also accompanies with complexities to manage private and public keys over the time. In addition to these implementations, the proposed architecture will employ on time stamp based access window to provide sharing facilities to multiple users upon standard cryptographic implementation.

# CHAPTER 2

# LITERATURE SURVEY

## 2.1 RELATED WORKS

### 2.1.1 IDENTITY BASED APPROACH

Identity Based Encryption Transformation for Flexible Sharing of Encrypted Data in Public Cloud (H. Deng et al., 2020) defines issues related to multi-tenant architecture where users need to access encrypted data stored in cloud environments. To handle this, data users are identified and authorized for data access based on their recognizable identities.

### 2.1.2 SECURE KEY MANAGEMENT

Diffie-Hellman Algorithm for Securing Medical Record Data Encryption Keys (Ermatita et al., 2020) work analyses the use of AES and Diffie-Hellman algorithms to encrypt and decrypt the medical images and analyses approaches to avoid data leakage to irresponsible parties.

### 2.1.3 INCREASED COMPUTATION EFFICIENCY

An Enhanced Cryptographic Algorithm in Securing Healthcare Medical Records (Jessie R. Paragas et al., 2020) presents idea of securing e-health records with modified Hill Cipher which analyses the unpredictability of the cipher-text as the known plaintext attack is handled with proposed methodology.

### 2.1.4 DISTRIBUTED STORAGE ISSUES

Secure Distributed Medical Record Storage using Blockchain (Sudarshan Parthasarathy et al., 2020) proposes the functionality of system to store Electronic records and techniques in smart contracts and encryption using random key.

### 2.1.5 QR CODE FLEXIBILITY

Applying QR Code to Secure Medical Management (Xuehu Yan et al., 2018) depicts possibilities of using secure QR code access to the patient’s medical records to improve medical management across organizations.

### 2.1.6 SEARCHABLE ENCRYPTION

Searchable encryption for healthcare clouds (R. Zhang et al., 2018) depicts approaches of searchable encryption in the domain of health care facilities. These approaches contrast and compare various searchable encryption schemes on accordance with their efficiency and real time secure performance.

### 2.1.7 SECURE DATA SHARING

Secure data sharing in cloud computing using revocable-storage identity-based encryption (J. Wei et al., 2018) depicts promising cryptographical primitive to build a practical data sharing system. Also, carries out performance comparisons in terms of functionality and efficiency to demonstrate its practicability.

## 2.2 SUMMARY OF THE LITERATURE SURVEY

Approach of Cascading algorithmic approach will improve computational costs, thereby increasing complexity for hackers and avoids attempts like dictionary attacks, hash table generation, etc… Discussions on algorithms to encrypt and decrypt voluminous medical data will also consider measures to avoid data leakage to irresponsible parties via secured communication channel establishment. Also key management strategies should be established to establish tamper proof records and quick access of those records are carried out with the help of QR code establishment with secure encoding of identity.

## CHAPTER 3

# SYSTEM ARCHITECTURE AND DESIGN

### 3.1 SYSTEM ARCHITECTURE

In this approach, a secure cloud-based application for Health care sector is proposed to manage the medical records and perform functionalities with it. The secure storage and retrieval of medical records is backed up with Blowfish algorithm and hashing will be involved to alter plaintext and improve resistance against attacks.

The cryptographic algorithms will allow the system to function efficiently and makes the approach cost effective, also scalability of the application will make it preferable for legacy systems. The confidentiality of reports will be taken care and memory efficiency and time for execution will play a major factor in deciding a strong cryptographic algorithm for larger resources. Cascading of algorithm will make the approach efficient against hackers and universal patient id will be useful for Quick Access of records.

The public key encryption process works only if the recipient has already decided to use it and has made a key available. And most system find it productive to establish management of keys. Utilisation of Private Key Generator encounters trust issues on the end-user sides. An efficient approach for establishing trust on employing an Hierarchical Trust model will ensure liabilities of PKG in Identity Based Encryption.

The proposed architecture encloses the below mentioned approaches:

- Encryption and Decryption parallelization in Blowfish module with the use of counter variable with cipher mode built based on block cipher scheme.
- Key factor computation in PKG side in TC - IBE sharing system, which provides resistance against replay attacks.

- Acceptance window is limited to get access to the shared identity in TC scheme.
- The key generation is carried out with dynamic salt computation with respect to users and employ time based access window.
- In python implementation, generators and iterators are used to manage memory efficiently with yield statements in modules.
- The module proposed does not require padding as of traditional schemas.
- Cascaded encoding scheme along with security module is carried out at endpoint routes.

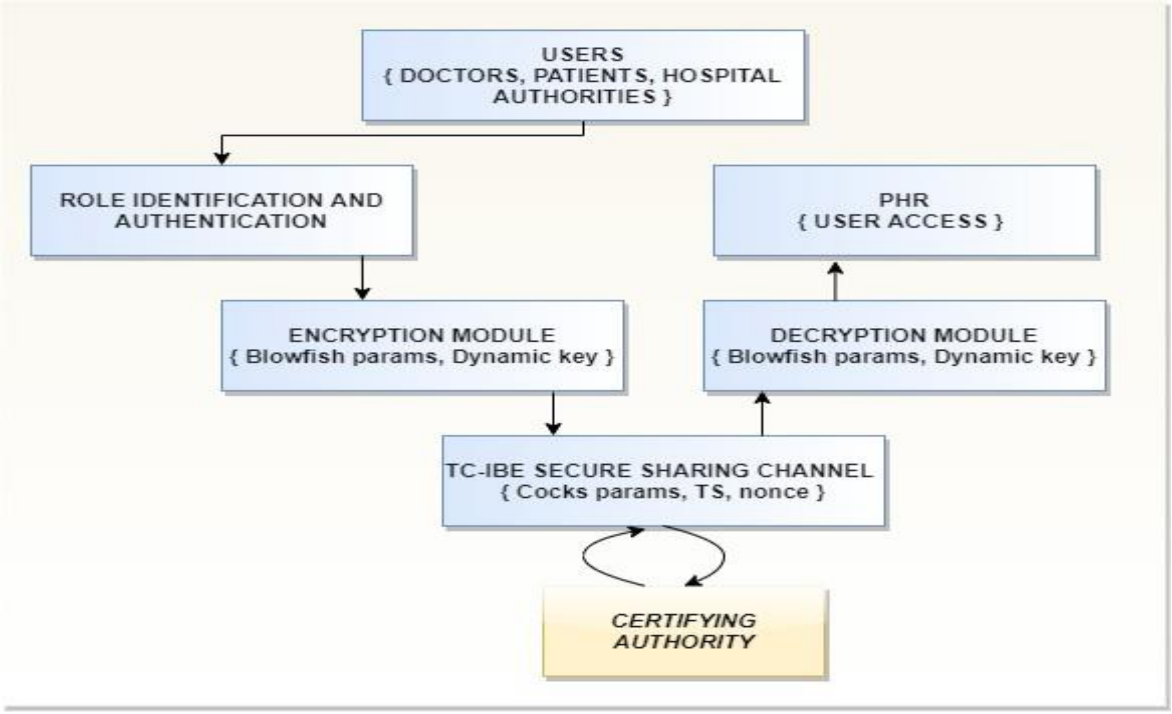

USERS
{ DOCTORS, PATIENTS, HOSPITAL AUTHORITIES }
ROLE IDENTIFICATION AND AUTHENTICATION
PHR
{ USER ACCESS }
ENCRYPTION MODULE
{ Blowfish params, Dynamic key }
DECRYPTION MODULE
{ Blowfish params, Dynamic key }
TC-IBE SECURE SHARING CHANNEL
{ Cocks params, TS, nonce }
CERTIFYING AUTHORITY

**Figure 3.1** – Block Diagram of Proposed System

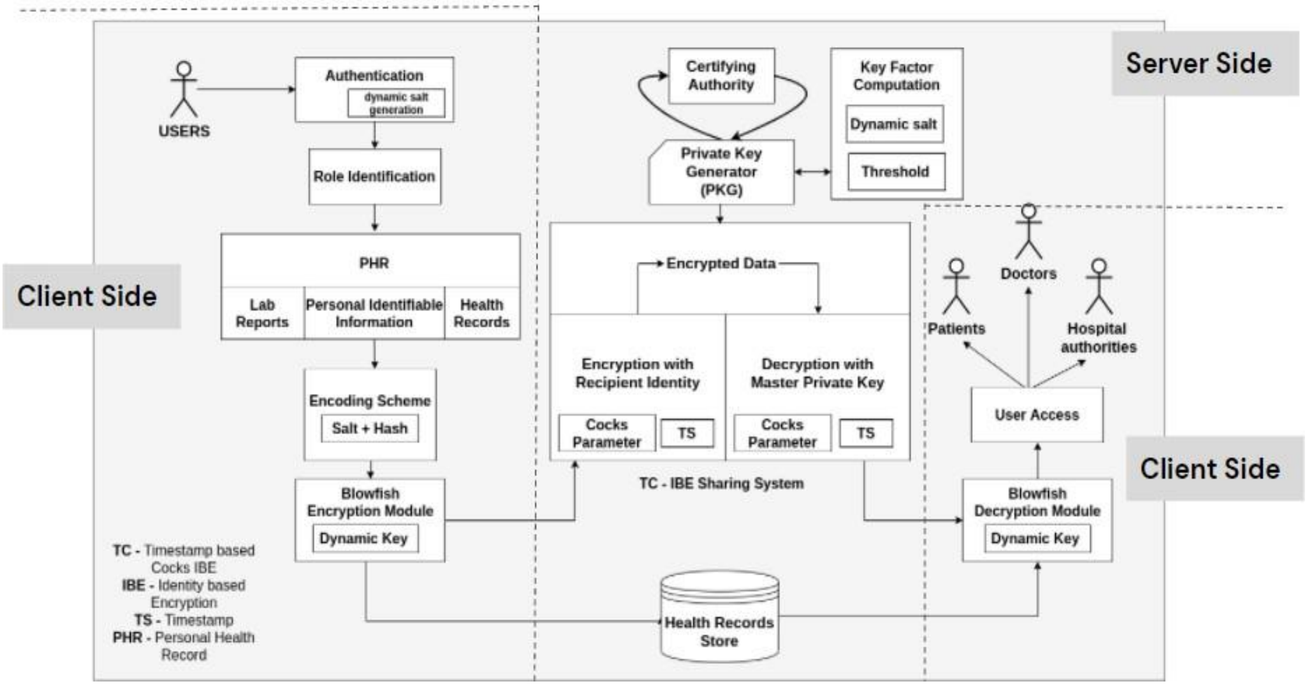


**Figure 3.2** – System Architecture

## 3.2 DETAILED DESIGN:

Here in the devised approach, a secure cloud-based application for Health care sector is proposed to manage the medical records and perform functionalities with it. The storage and sharing of medical records will employ Identity based encryption on the records with the usage of Blowfish algorithm in the underlying architecture and hashing will be involved to commute the summary of data and improve resistance against attacks. Salting is also carried out to protect attack against hash tables.

To emphasis on this proposed approach, the public key encryption process works only if the recipient has already decided to use it and has made a key available. And most people don't have public keys. So in this case the proposed

TC-IBE approach comes beneficial to use.

- Upon Patient registration, a unique hash will be generated with the help of Identity Generator module to store Personal Identifiable Information. Then, the patients’ health records will be uploaded to the system by the Hospital authorities restricted to their roles access.
- The role based access control model proposed will help us to maintain integrity among data as the high priority parties will get more access rights to the health records and the users like patients or insurance authorities will get only read access to the health data. The generated unique hash will be utilized to generate QR code access to the data in a secure fashion.
- The patients’ health records comprises of medical prescriptions, lab reports, diagnosis data, and other miscellaneous. These data are to be stored in a cloud database in a confidential manner with of utilizing Blowfish Encryption module proposed in the system.
- In modified Blowfish algorithm, to encrypt and decrypt data the well-known Feistal structure is employed in Counter based CTS mode of operation. Cipher text stealing (CTS) method of operation employ ciphers in blocks along with computation of plaintext messages which are not divisible in blocks and complexity is managed to encrypt uneven blocks. The modified blowfish algorithm operates on dynamic key, which highly influences the S-boxes in symmetric key algorithms. The substitution and permutation functions are carried out in configured rounds to arrive at cipher text with counter based approach. The random generated nonce is utilized along with counter variable to track rounds of encryption in Blowfish approach. Randomly generated keys are associated with users of the secure health record sharing facilities. Upon the primary authentication, these keys will be configured in database. The decryption

process performs the specified functions in reverse order with generated keys to decipher the ciphertext to arrive at appropriate messages.

- The important credentials for this algorithm and the related modules such as Secret key, environment setup credentials will be kept in .env file in server side. And in python implementation decouple module is utilized to read and write into the configuration files. And also separate modules are written for utility functions like converting string to bytes, bytes to string, base64 encoding and decoding and for salting and desalting, etc…
- Also upon carrying out sharing of records among various users/entities will be feasible through Identity Based Encryption Scheme proposed with the trusted Private key generator module, which will be verified with Certifying authority. The verification will compute checksum of the hashes of the certificate in regular interval as timestamp is maintained to compute hashes of the server authenticity.
- Upon sharing of medical records the sender will give access to his/her own data with employing IBE system encryption with recipient id and this in turn produce ciphertext of identity string and allows decryption module on the recipient side to obtain the master private key on accordance with his/her identity and decrypts the ciphertext to access the medical records data after successful decoding.
- The TC-IBE encryption scheme based Healthcare application is configured in Linux server in Amazon EC2 instance and facilitates health record storage in cloud Mongo database. And an Identity encryption supports transfer of file, and it will be downloaded to specified recipient, and it will get decrypted with the built-in proposed Blowfish encryption module.
- The key factor computation is facilitated with random nonce for each session and time stamp based access window is configured to avoid

replay attacks of attackers.

- A secure communication channel is configured with session id and access identity to share medical reports in established channel.
- At this point, the receiver who is authenticated with their session id in the configured time window access will able to invoke TC-IBE decryption module to access health records which are shared to their identities by the configured authorities of hospitals.
- In protocols that employ timestamp, the receiver will defines an acceptance window and will securely stores the received messages till time stamp expires.
- The acceptance window, the message delivery speed and frequency of access of records fetched by authorities will be monitored to facilitate quick response with further processing queries invoked by authorities and will able to deliver medical records in quick manner if the same identity is invoked at the second time.
- The identities associated with time bound will be able to access the shared records in the specified time. And after the expiry of time bound the records shared with the entities will expiry and the receivers will no longer able to read the medical records. They should once again prompt the sender to share the access to the receiver to view records for further suggestions on the shared records.
- If a malicious user is entering into authentication channel, the user identity is verified in the secure channel and without proper identity check the user access in invalidated and further the tampering of records is not possible thereby ensuring the integrity of the medical records.

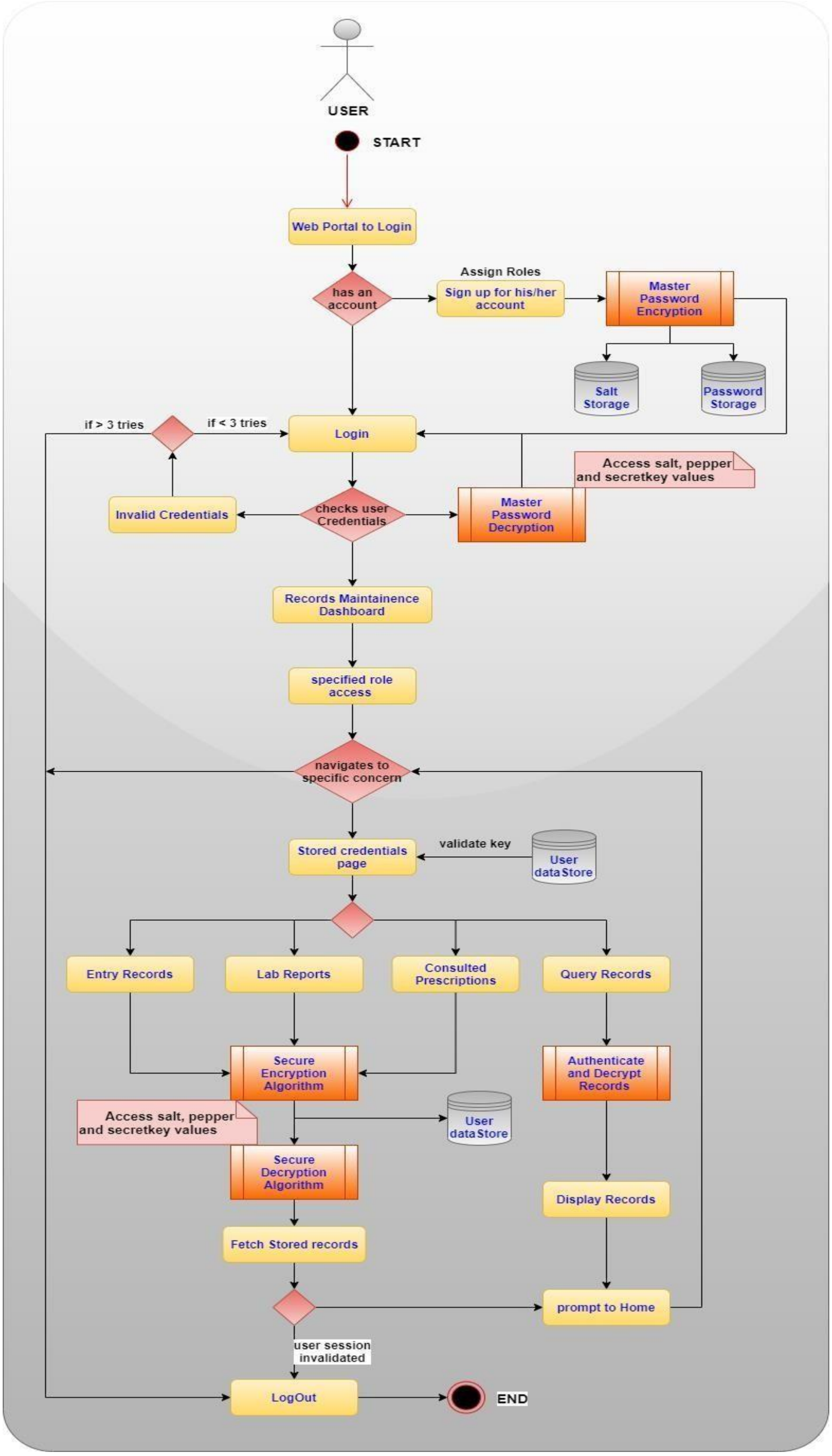


**Figure 3.3** – Flow of activities in Proposed System

- The timestamps are equivalent to sequence numbers. Furthermore, synchronization of sender and receiver clocks' is of a timeliness issue, and which constitutes interoperability issues.
- The proposed system will manage the recorded past medical records in the cloud instance of MongoDB data store.
- Redis database is utilised in proposed secure sharing architecture to manage faster record retrieval. Redis queries employs quicker response time, allowing the end users to fetch millions of user requests.
- These database supports complex query operations and as it is a in-memory cache it decreases data access latency and increases throughput. The cached data is made available in in-memory for persistant amount of time declared as variable in the store function, as the specified time expiries the data in cache gets cleared and the cache implementation starts monitoring the next frequent access identity to gets it to cache for quicker fetch operation.

## 3.3 HIPAA OBJECTIVES

As per the Health informatics, healthcare data breaches cost the industry approximately $5.6 billion every year. Being targeted by attackers as it possesses much information, private data, and financial information such as credit card details and information related to medical research and innovation.

The encryption technique discussed here along with access control will Confidentiality. So that Proposed System will comply with HIPAA and privacy rules.

The HIPAA legislation had four primary objectives:

- Control access and keeping data safe with appropriate security logins.
- Reducing fraud and deceit in healthcare.

- Enforcing high calibre for health information.
- Guarantee security and privacy of health information.

The proposed module satisfies the objectives quoted by Health Insurance Portability and Accountability Act (HIPAA) as below:

**Table 3.1** – Objectives met as per HIPAA

| **HIPAA OBJECTIVES** | **PROPOSED SYSTEM OBJECTIVES** |
|---|---|
| Control access and keeping data safe with appropriate security logins. | Role based access control system with proper authentication |
| Reduce healthcare fraud and abuse. | Tampering is avoided within proper encrypted channel |
| Guarantee security and privacy of health information. | TC-IBE system is proposed with dynamic key generation |
| Administrative simplification. | API Endpoints are defined with proper user console |

## 3.4 ROLE BASED ACCESS CONTROL

Role Based Access Control System is deployed to limit the access to the patients' health records information to valid authenticate users only. The access control is based on the hierarchy of the entities in the hospital infrastructure. The proposed solution take care of the security measurement on accordance with role based access control system. In the healthcare infrastructure, the authorities within premises will be automatically trusted and whenever each entity is allowed to communicate to the other entity, the entities must be verified before records sharing channel is established.

PHI's and PII's are dealt inside hospital authorities and their security requirement must be satisfied at all costs.

Each hospital authorities must be compactible to establish clear guidelines for role based access control systems and specific access control is implemented with of doctors, patients, and staffs.

The devised hospital access control system explicitly defines the authorization controls of every single authority bounded in the hospital premises. RBAC authorizes users of the established secure healthcare system and couples users with permissions by introducing roles.

$$U \in User\ \boldsymbol{has}\ P \in Permissios\boldsymbol{:}\ U\boldsymbol{,}\ P \in AC \qquad \textbf{(3.2)}$$

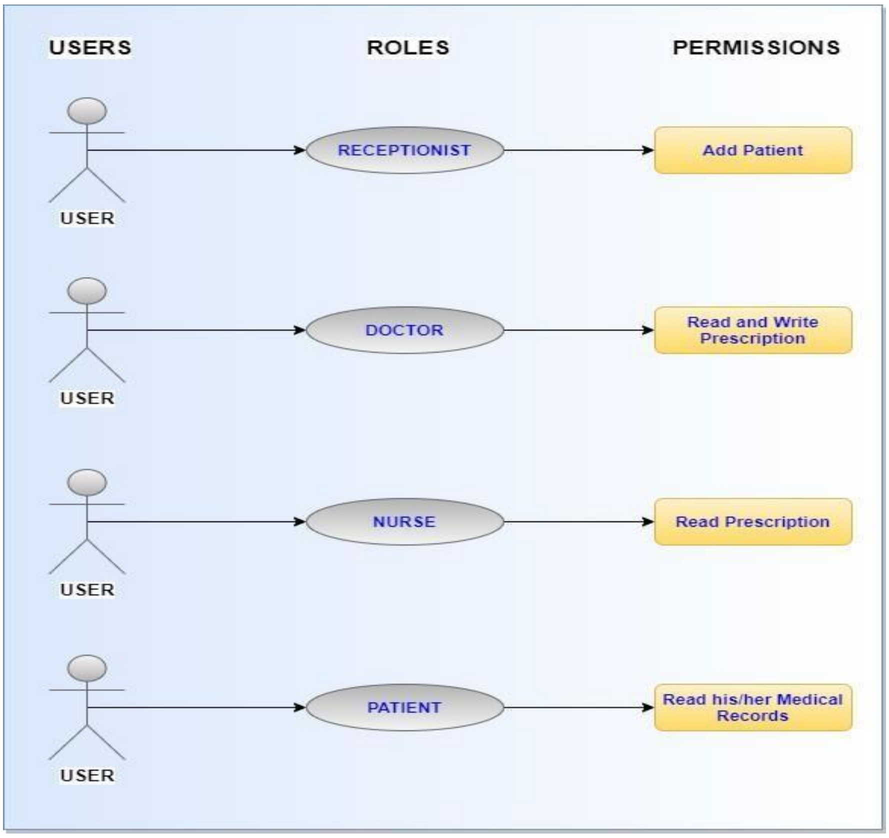


**Figure 3.4** – Relationship between Hospital entities and privilege

The hospital environment encompasses doctors, nurses, patients, administrators, receptionist, electronic health records, etc... Data composed of personal patient information and health history of patients will diversely affect the premises if these data are not landed in right hands and not handled appropriately. The security of these details are most valuable in healthcare sector applications and proposed health record sharing platform will enforce basic security norms. The allowance of access to the medical records among multiple users for specific purposes is handled appropriately. Records in hospitals is heavily regulated; due to the sensitive and privacy implications.

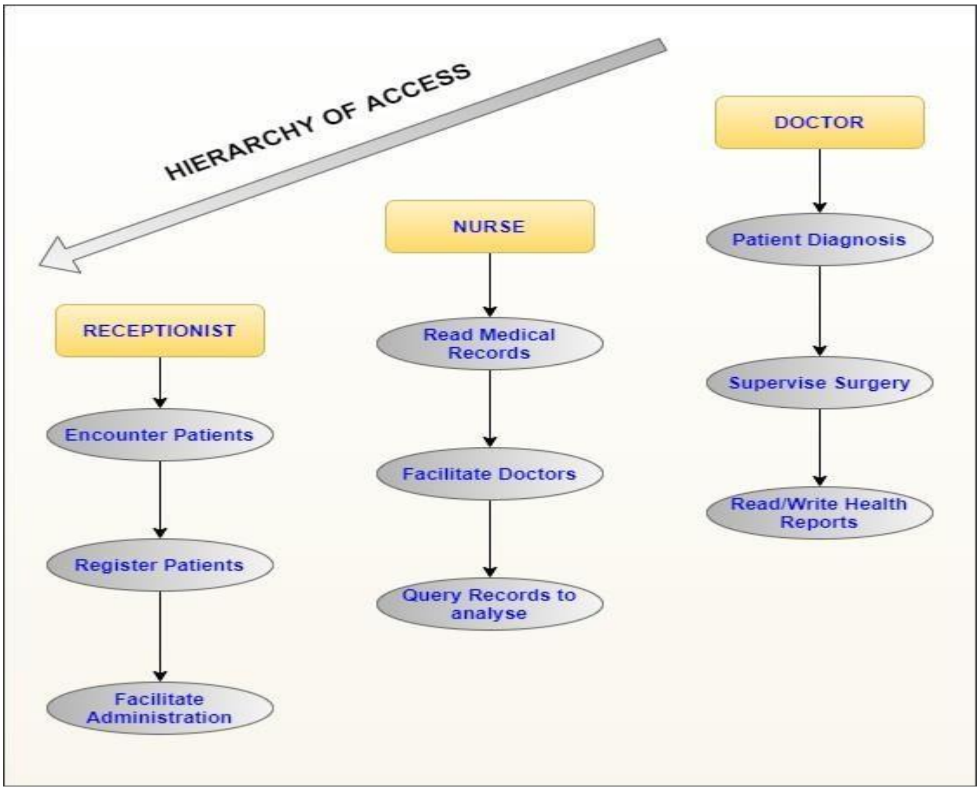


**Figure 3.5** – Hierarchy model in Hospital

Below are the defined terms in relation to the hospital environment role management policies:

**Administrator** : a person who has access to the secure data store of the hospital management and assigns role to the user and authorizes their entry.

**Doctor** : a person who has read and write access to the medical records of patients and modify the prescriptions, lab reports of the patients.

**Nurse** : a person who has control to read the prescriptions and has limitations in viewing the entire medical records of desired patients.

**Receptionist :** a person who registers the patients in the hospital premises and manages the personal identifiable information of the patients, but possess restrictions on viewing health records of patients.

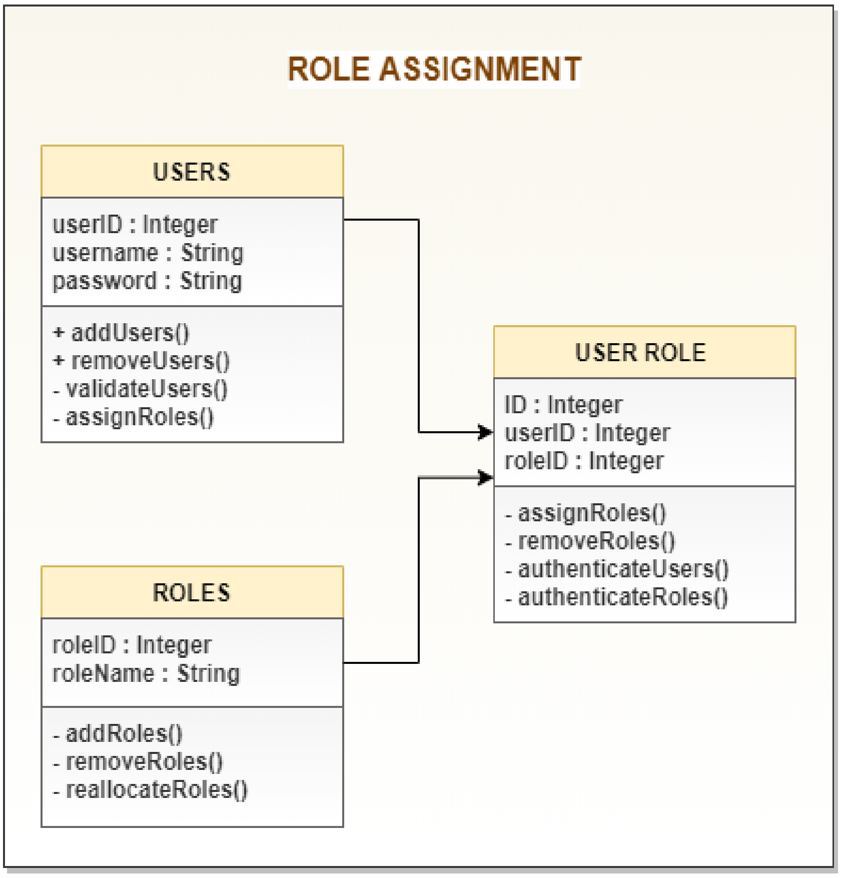


**Figure 3.6** – Role-Based Access Control

## 3.5 TRUST MODEL FOR PKG

For building a trust relationship among entities of health care, a trust model is devised. The third party Private Key Infrastructure will ensure trust relationship among hospital entities for establishment of secure communication channel.

1. Creation of Trust model.
2. Before the users/hospital authorities establish communication between them, they will check their authenticity by Certifying authorities.
3. Once the users are authenticated with PKG, the PKG's authority has no trust issues with the established user base and facilitate key

computation.

4. Separate data store is managed to store Authorized certificates for future references.
5. Monitoring and maintenance of PKG will be facilitated in convenient manner.

### 3.6 HEALTHCARE SECURITY REQUIREMENT

- Confidentiality: Patient information and medical data sent by emails, such as lab reports sent to a physician, emails from a doctor to another practitioner consulting for a second opinion, are private and highly sensitive.
- Solving the scarcity of resources: The doctors in urban health centers can be availed for guidance from rural environments.
- Regulation: Secure storage of patient information and medical record. Disclosure of PHIs' details will attract fines.
- Extended use of IT: To reduce financial hazards many health care facilities make use posting mails and engaging faxing to convey sensitive credentials which increased operational costs.
- Central data repository: The data store can be utilized as national medical research repository, disease control, and epidemics monitoring.

## CHAPTER 4

# ALGORITHM DEVELOPMENT AND IMPLEMENTATION

### 4.1 BLOWFISH ALGORITHM WITH DYNAMIC KEY

Blowfish utilizes same key to perform the encryption and decryption of plaintext, thereby popularly known as symmetric encryption algorithm. Blowfish falls under the category of block cipher as it divides the plaintext into fixed block sizes to carry out encryption and decryption on the messages.

To increase complexity, Blowfish utilizes the 64-bit block size thereby making the key secure. The proposed modified version of blowfish algorithm will systematically generates dynamic key on accordance with user thereby securing the implemented systems from the dictionary attacks.

Also, the plaintext which is fed by user is encoded with PEM scheme and salting is carried out with dynamic salts to entirely transform the plaintext before it is passed onto the encryption module. The dynamic salt generation is carried out in the authentication phase of the implemented system. And the generated key is of 32 bits to 448 bits. After that phase, the encryption with blowfish algorithm accompanied with counter based ciphertext stealing approach. Once the key expansion phase gets completed, the data encryption phase will be initiated. The key expansion in the proposed architecture will convert the dynamic key into sub-keys, which are further gets utilized by the function F of the Blowfish data encryption phase. Now, in data encryption phase, the algorithm will carry out with 16 rounds of Fiestal cipher with key dependent S-boxes. The intial bytes will subdivided and XORed with elements of P array to generate values P' and carried over to Transformation function(F) and further gets XORed with right sectioned bits of the message. These computations are carried out for 15 more rounds with successive values of P array to produce ciphertext. The P array and S array values are already computed based on the dynamic key generated for specific users. These 2d array values are kept confidential. And proper salting and encoding schemes are configured with proper interaction of the blowfish encryption and decryption algorithms accompanied with TC-IBE system invoked as per the user requirement. The discussed encryption phases and transformation function structure are depicted in the below figures 4.1 and 4.2.

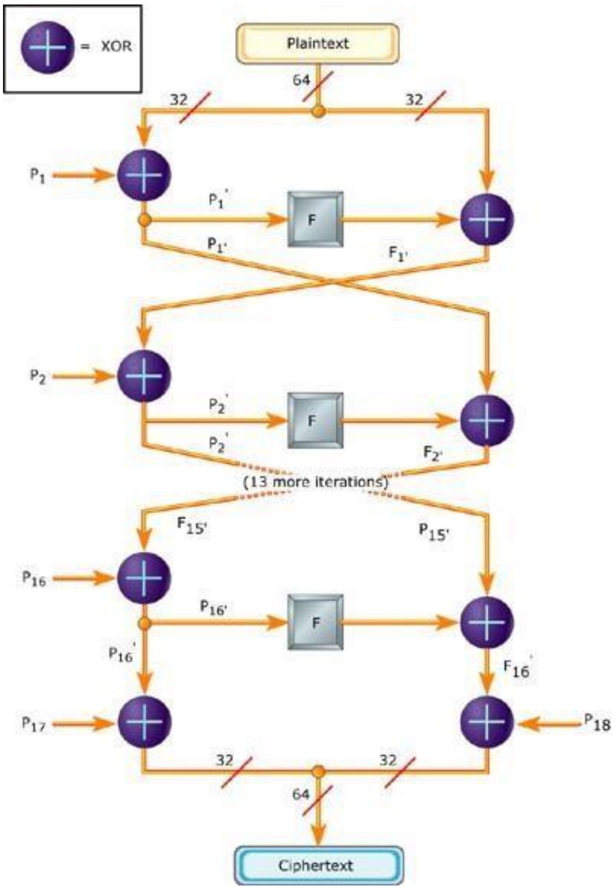


**Figure 4.1** – Blowfish Encryption

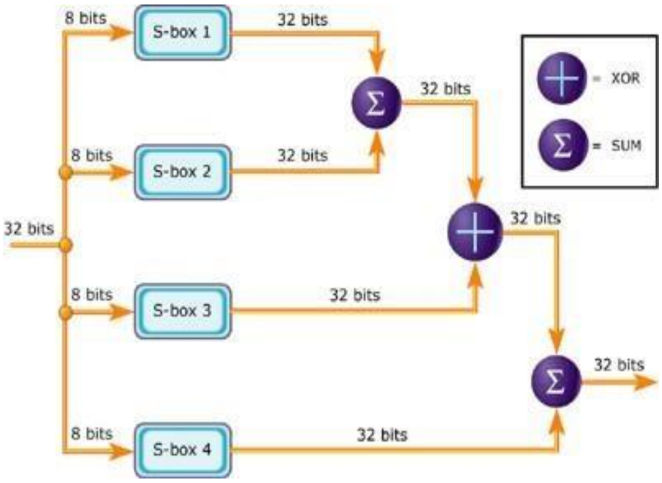


**Figure 4.2** – Function 'F' of Blowfish

### 4.1.1 PROPOSED COUNTER BASED BLOCK CIPHER

- No padding
- Can prepare encryption and decryption in advance
- Support for parallel computing
- Encryption and decryption uses the same block sizes

CTR uses Counter variable to encrypt and decrypt data in parallel on every time known counter is used. To ensure security, the key in this mode need to be changed for every 2^(n/2) encryption blocks.

- The padding can be eliminated with CTR and keystream generation for covering padding bits is eliminated.
- The decryption follows the same process as of encryption and the key stream is generated to perform XOR operation and plaintext is recovered.
- CTR function encrypt and decrypt other entities to decrypt the string at the starting of the function.
- Counter mode produces 16 byte block of keystream for encrypting the

plaintext and XOR function is carried out against plaintext.

### 4.1.2 CIPHERTEXT STEALING

**Encryption Mode**

1. $E_{n-1}$ = Encrypt ($K$, $P_{n-1}$). Encrypt $P_{n-1}$ to create $E_{n-1}$.
2. $C_n$ = Head ($E_{n-1}$, $M$). Select the first $M$ bits of $E_{n-1}$ to create $C_n$. The final ciphertext block, $C_n$, is composed of the leading $M$ bits of the second-to- last ciphertext block. In all cases, the last two blocks are sent in a different order than the corresponding plaintext blocks.
3. $D_n = P_n$ || Tail ($E_{n-1}$, $B{-}M$). Pad $P_n$ with the low order bits from $E_{n-1}$.
4. $C_{n-1}$ = Encrypt ($K$, $D_n$). Encrypt $D_n$ to create $C_{n-1}$. For the first $M$ bits, this is equivalent to what would happen in ECB mode (other than the ciphertext ordering). For the last $B{-}M$ bits, this is the second time that these data have been encrypted under this key.

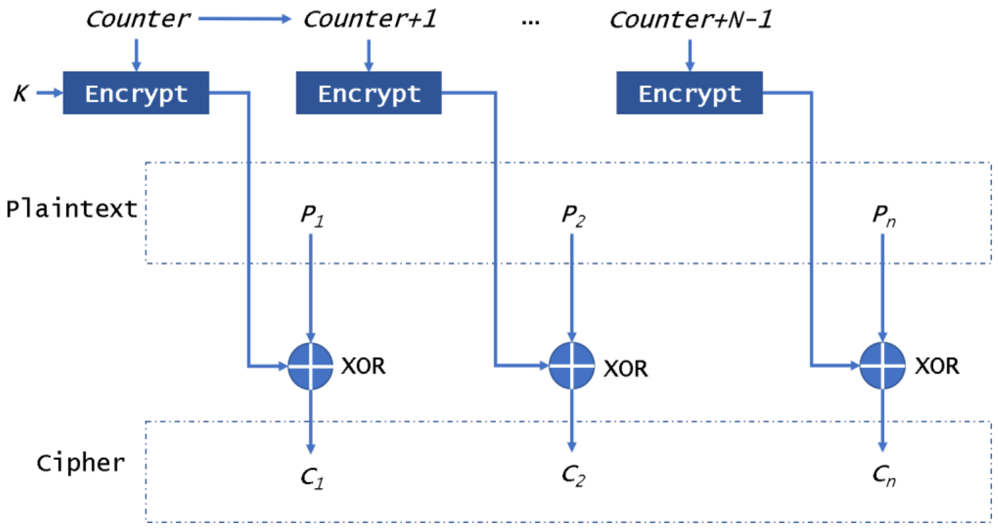


**Figure 4.3** – Encryption Mode

### Decryption Mode

1. $D_n$ = Decrypt ($K$, $C_{n-1}$). Decrypt $C_{n-1}$ to create $D_n$. This undoes step 4 of the encryption process.

2. $E_{n-1} = C_n \parallel$ Tail ($D_n$, $B{-}M$). Pad $C_n$ with the extracted ciphertext in the tail end of $D_n$

3. $P_n$ = Head ($D_n$, $M$). Select the first $M$ bits of $D_n$ to create $P_n$. The first $M$ bits of $D_n$ contain $P_n$. We queue this last (possibly partial) block for eventual output.

4. $P_{n-1}$ = Decrypt ($K$, $E_{n-1}$). Decrypt $E_{n-1}$ to create $P_{n-1}$. This reverses encryption step 1.

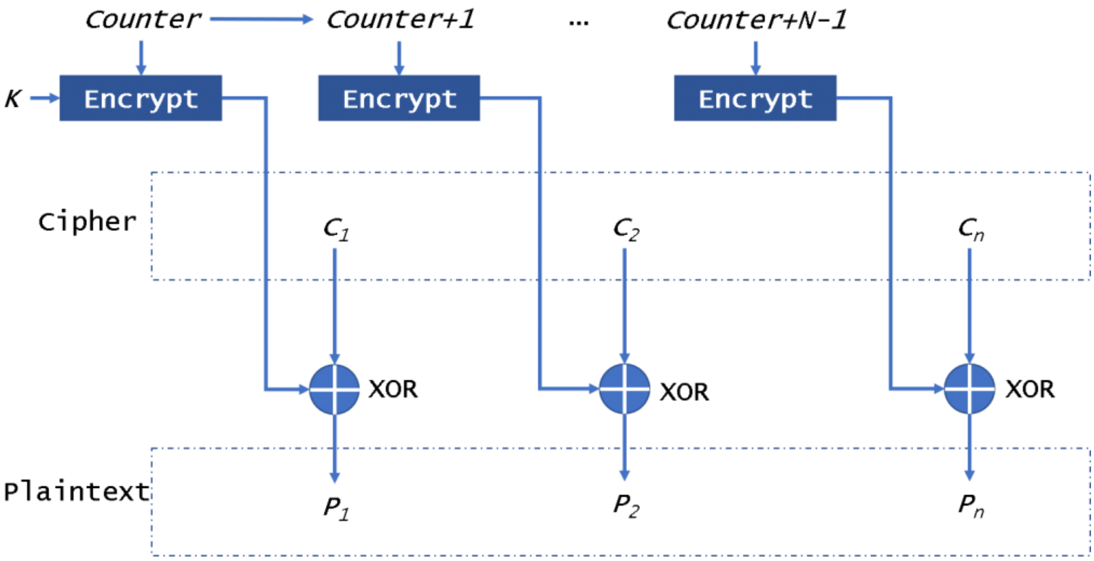


**Figure 4.4** – Decryption Mode

**Table 4.1** – Acronym Description

| SYMBOLS | DESCRIPTION | SYMBOLS | DESCRIPTION |
|---|---|---|---|
| $P_{n-1}$, $P_n$ | last two blocks of the plaintext | $K$ | key |
| $B$ | length of $P_{n-1}$ | $C_n$ | ciphertext |
| $M$ | length of $P_n$ | $E_{n-1}$, $E_n$ | encrypted text |

## 4.2 IDENTITY BASED ENCRYPTION

Identity-based encryption is a public key encryption in which a user's known unique identity is utilized to generate a public key and private key generator is employed to calculate a private key. There is no need to public keys ahead of communicating for sharing data among users.

The following four stages of algorithm will be part of an identity-based encryption scheme: Setup, Extract, Encrypt and Decrypt.

### SETUP

The PKG chooses:

1. A public RSA-modulus $\boldsymbol{n = pq,}$ where $\boldsymbol{p, q, p \equiv q \equiv 3 \ mod \ 4}$ are prime and kept secret,
2. The message and the cipher space $K = \{\boldsymbol{-1, 1}\}$, C= $\mathbb{Z}_n$ and
3. A secure public hash function $\boldsymbol{f} : \{\boldsymbol{0, 1}\} * \rightarrow \mathbb{Z}_n$.

### EXTRACT

The user obtains his/her private key by communicating with PKG after successful validation of user identity and thereby PKG is ensured by trust model for proper key factor computation. The PKG

1. derives $\boldsymbol{\alpha}$ with $(\frac{\alpha}{n}) = \boldsymbol{1}$ by a deterministic process from $\boldsymbol{ID}$

(e.g. multiple application of $\boldsymbol{f}$).

2. Computes $\boldsymbol{r} = \boldsymbol{a}^{(\boldsymbol{n}+5-\boldsymbol{p}-\boldsymbol{q})/8}$ $(\boldsymbol{mod\ n})$ ( which fulfils either $r^2 = a\ (mod\ n)\ or\ r^2 = -a(mod\ n)$,see below ) and

3.transmits **r** to the user.

## ENCRYPT

To encrypt a bit (coded as 1/-1) $\boldsymbol{m}\ \boldsymbol{\mathcal{E}}\ K$ for $ID$, the user

1. Chooses random $t_1$ with $m = (\frac{t_1}{n})$ ,
2. Chooses random $t_2$ with $m = (\frac{t_2}{n})$ , different from $t_1$ ,
3. Computes $c_1 = t_1 + at_1^{-1}\ (mod\ n)$ and $c_2 =\ t_2 - at_2^{-1}(mod\ n)$ and
4. Sends $s = (c_1\ , c_2)$ to the user.

## DECRYPT

To decipher a ciphertext $s = (c_1, c_2)$ for user ID ,he

1. Computes $a =\ \boldsymbol{c_1\ +\ 2r\ \ if\ r^2 = a\ or\ a = c_2 + 2r}$ otherwise, and
2. Computes $\boldsymbol{m} = (\frac{a}{\boldsymbol{n}})$.

**Table 4.2** – Acronym Description

| SYMBOLS | DESCRIPTION | SYMBOLS | DESCRIPTION |
|---|---|---|---|
| $p,q$ | prime numbers | $a$ | quadratic residue |
| $\boldsymbol{c_1, c_2}$ | Ciphertext | $t_1$ , $t_2$ | random variable |
| $m$ | Message | $f$ | hash function |

This proposed framework can be initiated by the sender and a unique identifier of the recipient is used to calculate a public key. A third party server namely

Private Key Generator (PKG) will generate appropriate private key for authenticated users from utilizing the public key of the receiver. The recipient can get access to their private keys from trusted PKG and the receiver can be free from worries of pre communicating of public keys. The sender can share access to the receiver with public key encrypted data and private key generator after identifying the receiver will generate private key with of configured trusted third party server and therefore it facilitates the sharing of records by decrypting shared data with private key.

On utilizing PKG, for every unique identity the PKG will generate private key with identity ID of the authenticated user. The sender will send the encrypted messages with of utilizing the Identity string of the receiver as public key and it gets decrypted and validated with of utilizing private key.

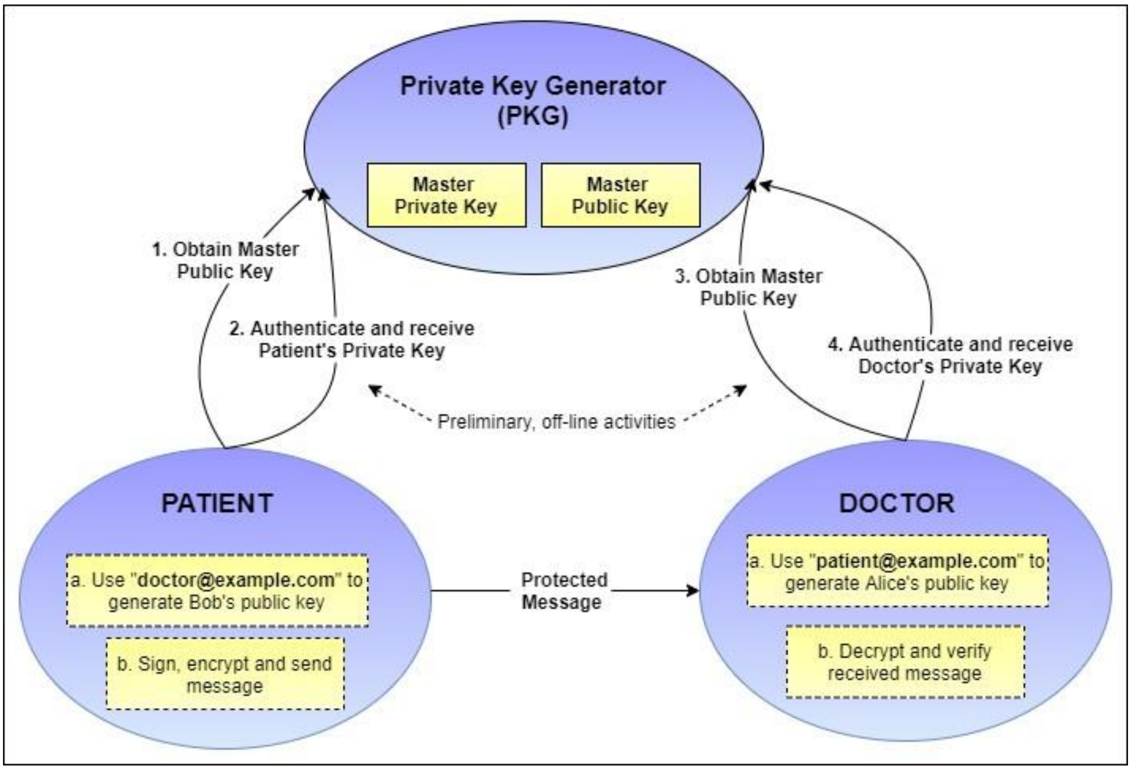

Private Key Generator (PKG)
Master Private Key
Master Public Key
1. Obtain Master Public Key
2. Authenticate and receive Patient's Private Key
3. Obtain Master Public Key
4. Authenticate and receive Doctor's Private Key
Preliminary, off-line activities
PATIENT
a. Use "doctor@example.com" to generate Bob's public key
b. Sign, encrypt and send message
Protected Message
DOCTOR
a. Use "patient@example.com" to generate Alice's public key
b. Decrypt and verify received message

**Figure 4.5** – IBE System Approach

## 4.3 PROPOSED ENCODING SCHEME

**STEP 1:**

- { S1, S2 } computed with ORG ID/USER ID
- PEM scheme (Privacy-enhanced Electronic Mail) is applied to every { P }
- Salting is done with proposed approach and feed into encrypt_bf function

**STEP 2:**

- Counter value is assigned with K to perform encryption, decryption module
- Hex_Encoder is utilized with modified plaintext if needed

**STEP 3:**

- When sharing system is invoked, ID is mapped along with TS and nonce
- IBE is utilized with generated KF and params { r, a } are serialized into pickle model with desired class object in PKG

**STEP 4:**

- Iterators and generators are utilized to reduce wastage of RAM space
- Cache is implemented in accordance frequent access id of records

**Table 4.3** – Acronym Description

| **SYMBOLS** | DESCRIPTION | **SYMBOLS** | DESCRIPTION |
|---|---|---|---|
| **S1, S2** | **Generated Dynamic Salt** | **P** | **Plaintext** |
| **K** | Dynamic key | **r, a** | PKG generated params |
| **KF** | Key Factor | **TS** | Time Stamp |

Proposed TC-IBE system is implemented with timestamp and nonce for carrying out record encryption. Unique id is generated with combination of bcrypt hashing and salting and the generated Id is associated with QR access code for quick retrieval. Time based access window is set to sharing credentials among hospital authorities in proposed TC-IBE system.

Cache is implemented with in-memory database for quicker access with the key value pairs associated with frequently accessing records. Access control is limited with api endpoint routes along with the proposed role based system.

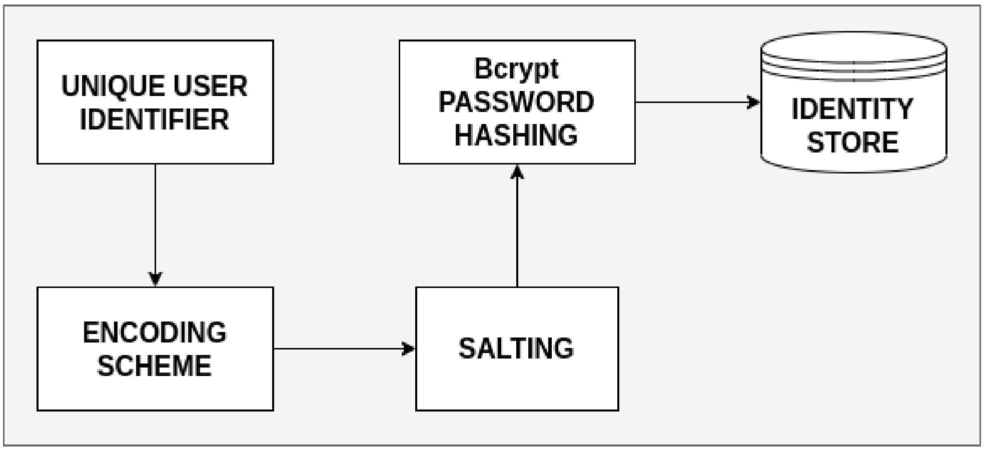


**Figure 4.6** – Identity Generator with proposed Encoding

## 4.4 SYSTEM IMPLEMENTATION

- Encryption and Decryption parallelization in Blowfish module with the use of counter variable with cipher mode built based on block cipher scheme.
- Key factor computation in PKG side in TC - IBE sharing system, which provides resistance against replay attacks.
- Acceptance window is limited to get access to the shared identity in TC scheme.
- The key generation is carried out with dynamic salt computation with respect to users and employ time based access window.
- In python implementation, generators and iterators are used to manage memory efficiently with yield statements in modules.
- The module proposed does not require padding as of traditional schemas. Cascaded encoding scheme along with security module is carried out at endpoint routes. Key Generation for Identity based approach is carried out with TimeStamp(TS) - These protocols will counteract replay and interleaving attacks, and to provide uniqueness or timeliness guarantees. Arbitrary numbers like Nonce are used for initialization vectors for Key generation - nonces within the relevant scope will make system more complicated, because it needs to maintain state reliably while scope is "alive".
- Dynamic salting with Tokenization is carried out in primary level of plaintext response - strengthen security, protect against dictionary attacks, brute-force attacks.

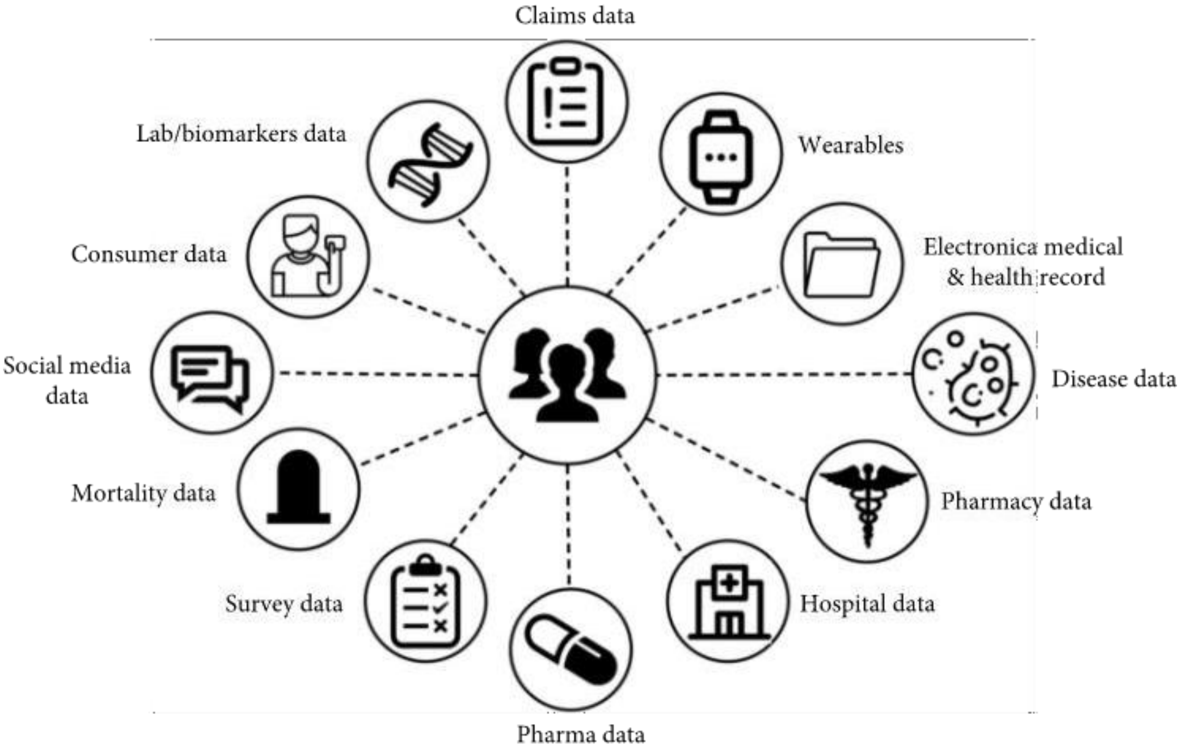


**Figure 4.7** – Healthdata services associated with the system

## 4.5 IMPLEMENTATION ENVIRONMENT

Python programming environment is used for implementation purpose, since it runs on all browsers and is platform independent and it is compactable with various server scripts. For data store, MongoDB is used. The client and server were implemented in Windows 10 (64 bit operating system). The Client interface is designed with Flask – a micro web framework written in python. The decouple module is utilized to read and configure environment and session variables.

# CHAPTER 5

# RESULTS AND DISCUSSIONS

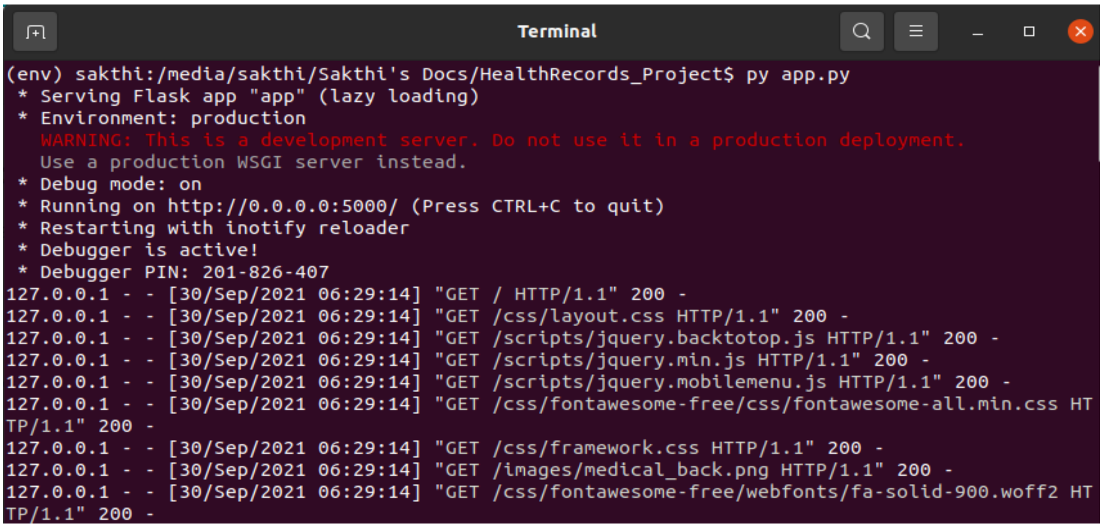
## 5.1 IMPLEMENTATION ENVIRONMENT

The client application console is designed with Flask web framework in Linux environment. The user authentication module is initiated with written python scripts, which involves dynamic salt generation. The secret credentials are managed in configuration files in server side and Pymongo module is utilized in python environment to connect to MongoDB instances which configures collections for health record storage and dynamic salt values. The database cluster handles the role based access control system with defined privilege values.

And the encryption and decryption modules are written in different files with predefined function calls. The database execution environment is facilitated with Redis cache with frequent accessing record storage to improve performance to fetch operations in database. OpenCV is utilized to operate with camera devices to scan Quick Response(QR) codes generated with unique identity string encoded in it for quicker and seamlessly health record retrieval. The Numpy library in python is handling the rendering the rendering of QR code data in matrix schemes. The overall system is hosted in AWS EC2 instance to facilitate global access to the secure identity based encryption and sharing channel.

## 5.2 EXECUTED OUTCOMES

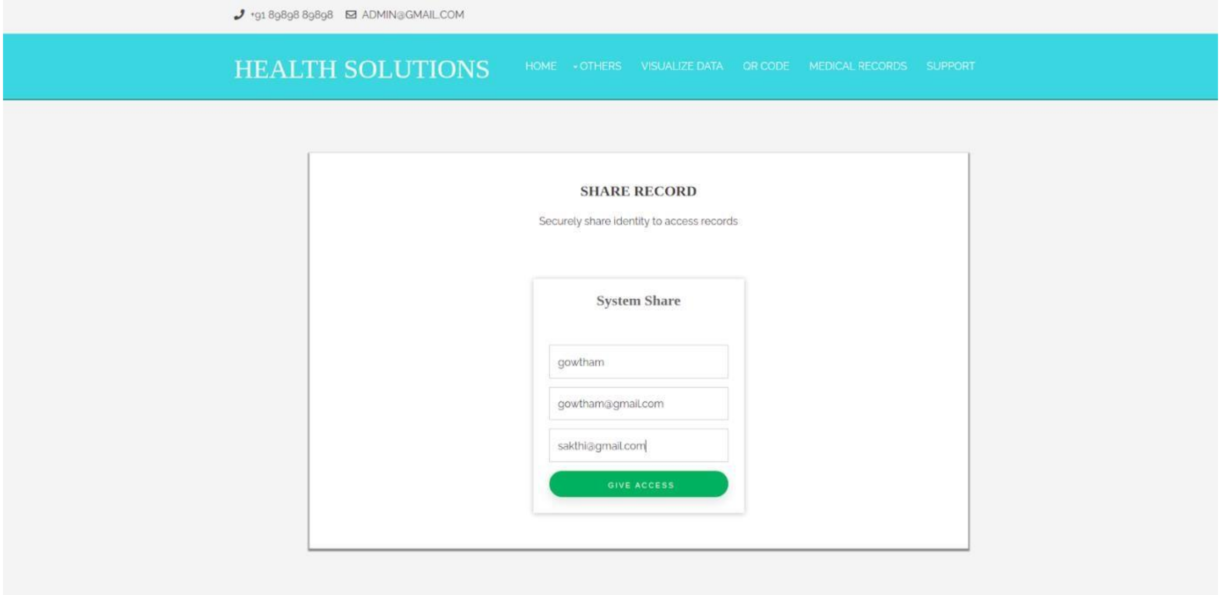
The proposed system is configured as Flask application and its execution is carried out in production WSGI server environment with debug mode on. The implemented secure Health records application is executed as shown in Figure 5.1.



```
(env) sakthi:/media/sakthi/Sakthi's Docs/HealthRecords_Project$ py app.py
 * Serving Flask app "app" (lazy loading)
 * Environment: production
   WARNING: This is a development server. Do not use it in a production deployment.
   Use a production WSGI server instead.
 * Debug mode: on
 * Running on http://0.0.0.0:5000/ (Press CTRL+C to quit)
 * Restarting with inotify reloader
 * Debugger is active!
 * Debugger PIN: 201-826-407
127.0.0.1 - - [30/Sep/2021 06:29:14] "GET / HTTP/1.1" 200 -
127.0.0.1 - - [30/Sep/2021 06:29:14] "GET /css/layout.css HTTP/1.1" 200 -
127.0.0.1 - - [30/Sep/2021 06:29:14] "GET /scripts/jquery.backtotop.js HTTP/1.1" 200 -
127.0.0.1 - - [30/Sep/2021 06:29:14] "GET /scripts/jquery.min.js HTTP/1.1" 200 -
127.0.0.1 - - [30/Sep/2021 06:29:14] "GET /scripts/jquery.mobilemenu.js HTTP/1.1" 200 -
127.0.0.1 - - [30/Sep/2021 06:29:14] "GET /css/fontawesome-free/css/fontawesome-all.min.css HT
TP/1.1" 200 -
127.0.0.1 - - [30/Sep/2021 06:29:14] "GET /css/framework.css HTTP/1.1" 200 -
127.0.0.1 - - [30/Sep/2021 06:29:14] "GET /images/medical_back.png HTTP/1.1" 200 -
127.0.0.1 - - [30/Sep/2021 06:29:14] "GET /css/fontawesome-free/webfonts/fa-solid-900.woff2 HT
TP/1.1" 200 -
```

**Figure 5.1** – Execution of Scripts

The web console characterized with fields for carrying out secure Timestamp-based Cocks Identity Based Encryption involving nonce and parameters r, a is depicted in the below Figure 5.2.



SHARE RECORD

Securely share identity to access records

System Share

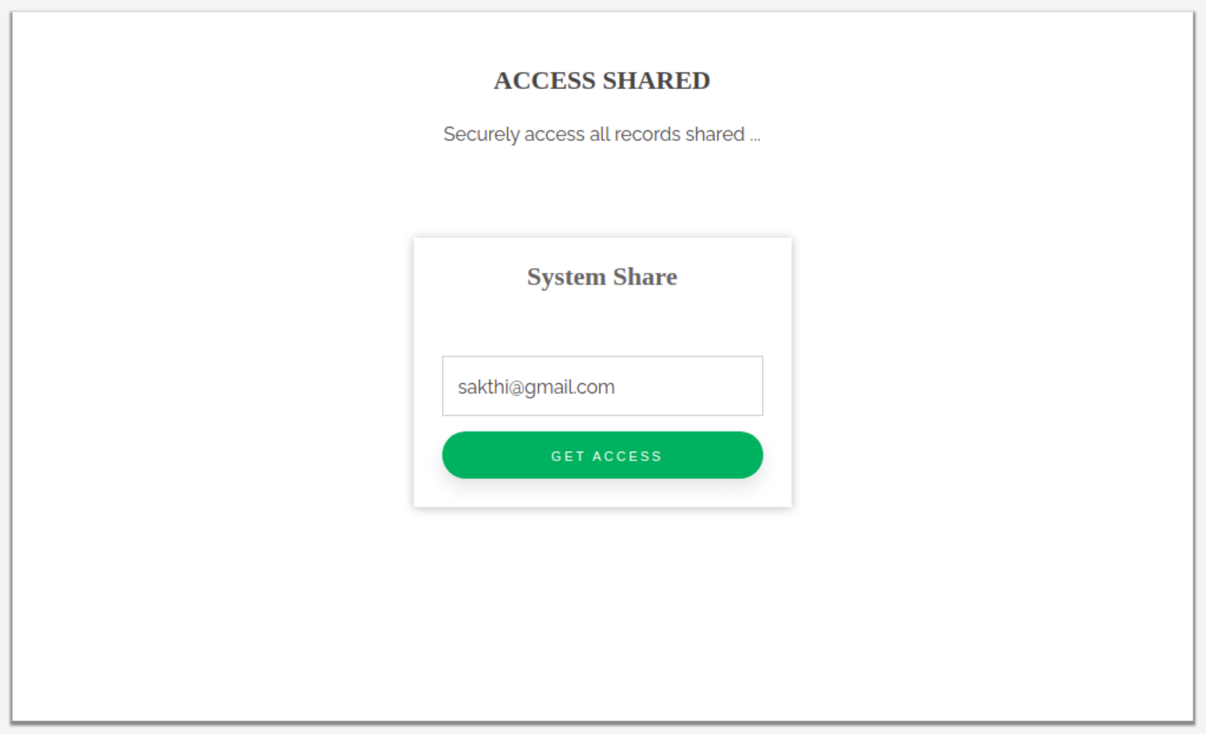
gowtham

gowtham@gmail.com

sakthi@gmail.com

**Figure 5.2** – TC-IBE Interaction Module

The secure Identity generator module is invoked after the primary authentication module on successful user sign in and within the allowed timestamp if the record has been shared with concerned entity, then their access is facilitated in Access Shared page as depicted in Figure 5.3.



ACCESS SHARED

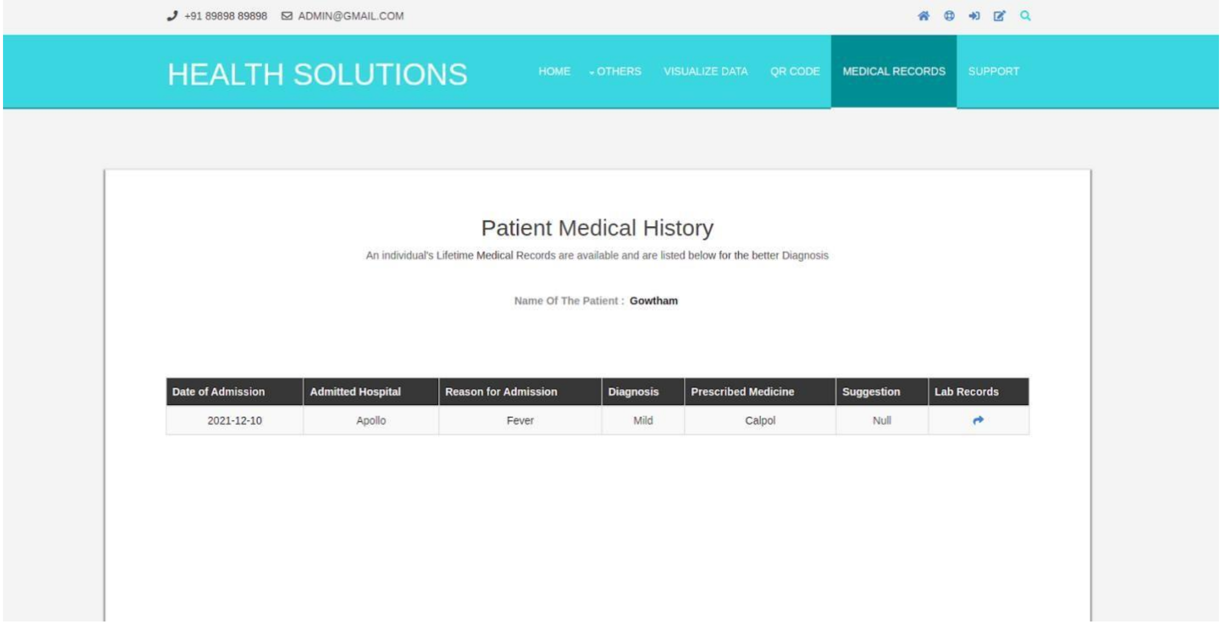
Securely access all records shared ...

System Share

sakthi@gmail.com

**Figure 5.3** – Invoke Shared Access

The encryption and decryption modules are defined in the functions encrypt_bf and decrypt_bf with of utilizing the counter based Blowfish algorithm with improved Key generation system facilitated with Dynamic key generation with respect to users. And Private Key Generator employed in IBE system is supported by key factor computation involving nonce (number once) and within allowed time bound if access is invoked then shared patients' health record is tabulated as shown in Figure 5.4.



Patient Medical History

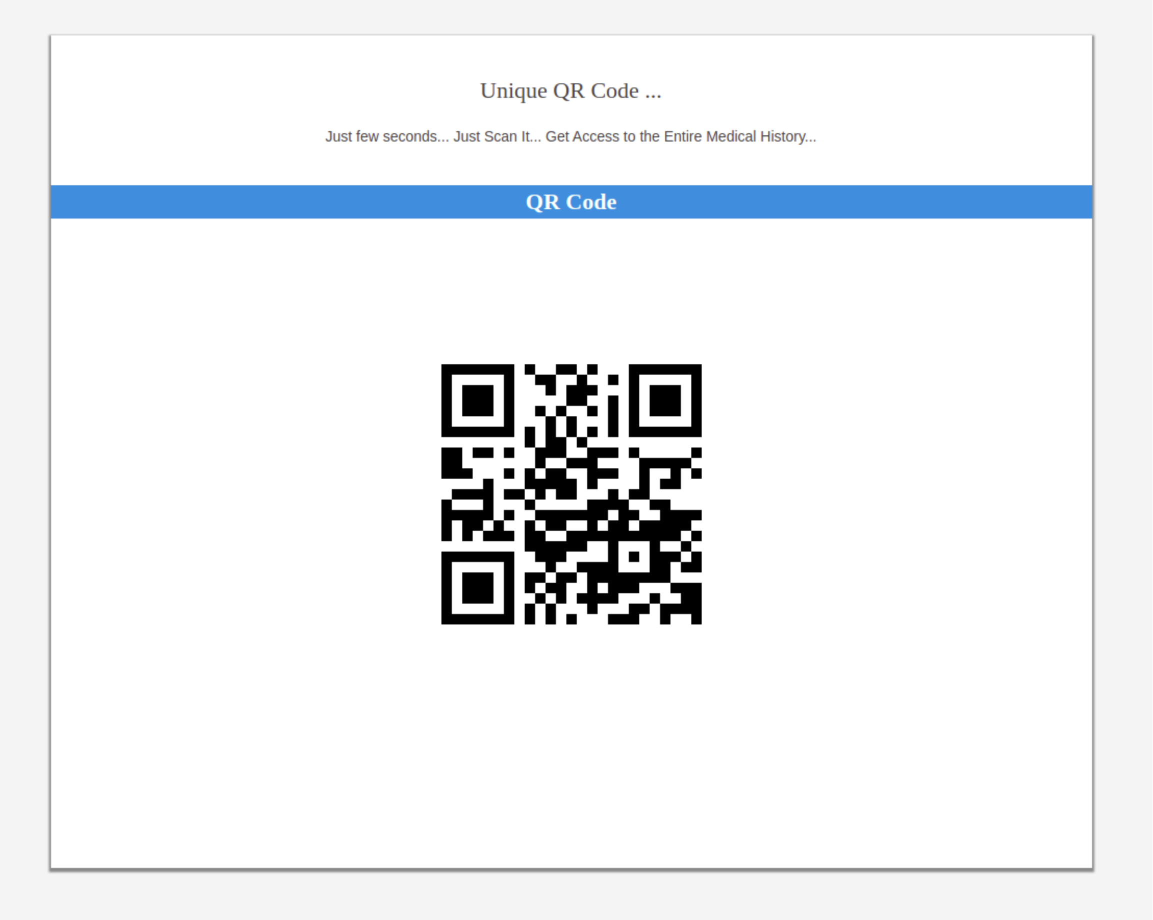
An individual's Lifetime Medical Records are available and are listed below for the better Diagnosis

Name Of The Patient : **Gowtham**

| Date of Admission | Admitted Hospital | Reason for Admission | Diagnosis | Prescribed Medicine | Suggestion | Lab Records |
|---|---|---|---|---|---|---|
| 2021-12-10 | Apollo | Fever | Mild | Calpol | Null | ↪ |

**Figure 5.4** – Fetched Health Records after Decryption

The Quick Response Code is generated by encoded Health Records Identity string. The quick access of records will employ QR codes to get access to patients’ past health records to facilitate emergency services. A medical QR card will contain information regarding to the patient's past health history, diagnosed prescriptions, medication details of the patients and lab reports. Quick Response (QR) codes will securely encode the desired identity string of the patients’ health system. And thereby patient health care records can be facilitated inside hospital authorities for quicker treatment access. The generated QR code for past Health Records is depicted in below Figure 5.5.

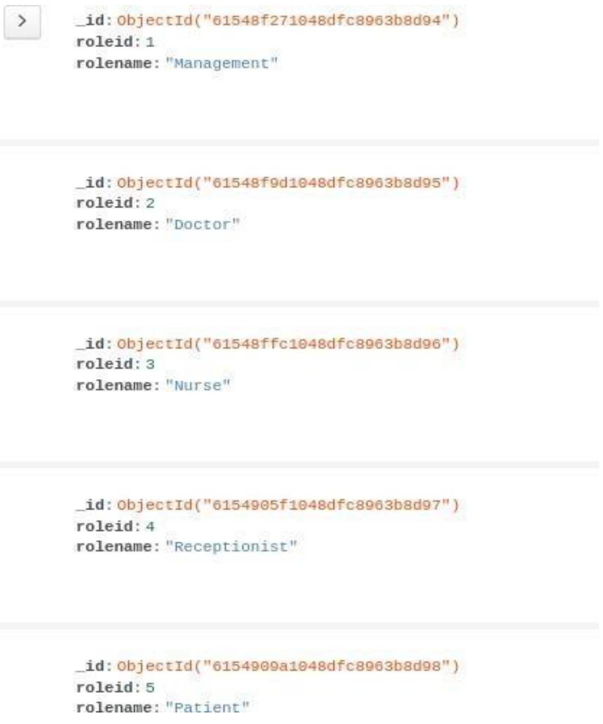


Unique QR Code ...

Just few seconds... Just Scan It... Get Access to the Entire Medical History...

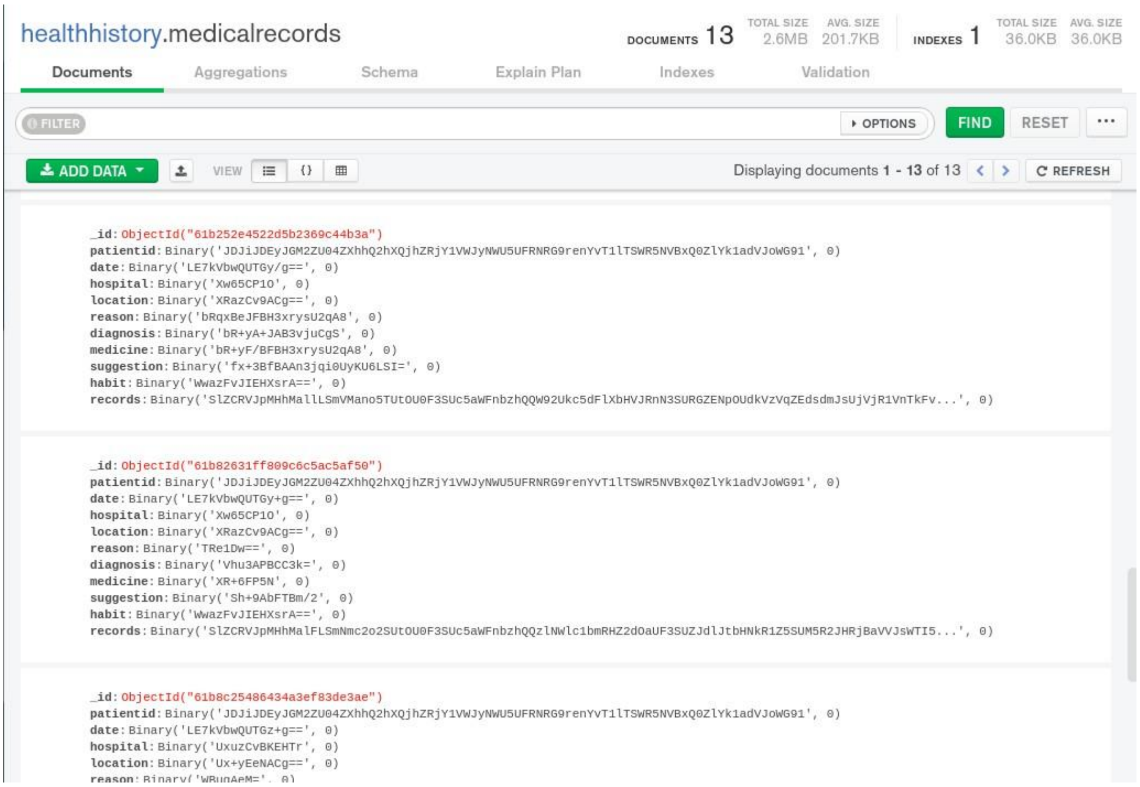
**Figure 5.5** – Generated QR Code for Health Records

The role based access control (RBAC) system is to govern access to health record sharing system. A user is granted one or more roles that determine the user's access to database resources and operations. A role grants privileges to perform the specified actions. The RBAC system is backed up with python implementation in backend to facilitate access to the end users. The encrypted data gets stored in MongoDB after computation from Modified Blowfish and TC-IBE system. The RBAC and DataStore are depicted in below Figure 5.6 and Figure 5.7.



```
_id: ObjectId("61548f271048dfc8963b8d94")
roleid: 1
rolename: "Management"

_id: ObjectId("61548f9d1048dfc8963b8d95")
roleid: 2
rolename: "Doctor"

_id: ObjectId("61548ffc1048dfc8963b8d96")
roleid: 3
rolename: "Nurse"

_id: ObjectId("6154905f1048dfc8963b8d97")
roleid: 4
rolename: "Receptionist"

_id: ObjectId("6154909a1048dfc8963b8d98")
roleid: 5
rolename: "Patient"
```

**Figure 5.6 –** Role Based Access Control System



```
_id: ObjectId("61b252e4522d5b2369c44b3a")
patientid: Binary('JDJiJDEyJGM2ZU04ZXhhQ2hXQjhZRjY1VWJyNWU5UFRNRG9renYvT1lTSWR5NVBxQ0ZlYk1adVJoWG91', 0)
date: Binary('LE7kVbwQUTGy/g==', 0)
hospital: Binary('Xw65CP1O', 0)
location: Binary('XRazCv9ACg==', 0)
reason: Binary('bRqxBeJFBH3xrysU2qA8', 0)
diagnosis: Binary('bR+yA+JAB3vjuCgS', 0)
medicine: Binary('bR+yF/BFBH3xrysU2qA8', 0)
suggestion: Binary('fx+3BfBAAn3jqi0UyKU6LSI=', 0)
habit: Binary('WwazFvJIEHXsrA==', 0)
records: Binary('SlZCRVJpMHhMallLSmVMano5TUtOU0F3SUc5aWFnbzhQQW92Ukc5dFlXbHVJRnN3SURGZENpOUdkVzVqZEdsdmJsUjVjR1VnTkFv...', 0)

_id: ObjectId("61b82631ff809c6c5ac5af50")
patientid: Binary('JDJiJDEyJGM2ZU04ZXhhQ2hXQjhZRjY1VWJyNWU5UFRNRG9renYvT1lTSWR5NVBxQ0ZlYk1adVJoWG91', 0)
date: Binary('LE7kVbwQUTGy+g==', 0)
hospital: Binary('Xw65CP1O', 0)
location: Binary('XRazCv9ACg==', 0)
reason: Binary('TRe1Dw==', 0)
diagnosis: Binary('Vhu3APBCC3k=', 0)
medicine: Binary('XR+6FP5N', 0)
suggestion: Binary('Sh+9AbFTBm/2', 0)
habit: Binary('WwazFvJIEHXsrA==', 0)
records: Binary('SlZCRVJpMHhMalFLSmNmc2o2SUtOU0F3SUc5aWFnbzhQQzlNWlc1bmRHZ2dOaUF3SUZJdlJtbHNkR1Z5SUM5R2JHRjBaVVJsWTI5...', 0)

_id: ObjectId("61b8c25486434a3ef83de3ae")
patientid: Binary('JDJiJDEyJGM2ZU04ZXhhQ2hXQjhZRjY1VWJyNWU5UFRNRG9renYvT1lTSWR5NVBxQ0ZlYk1adVJoWG91', 0)
date: Binary('LE7kVbwQUTGz+g==', 0)
hospital: Binary('UxuzCvBKEHTr', 0)
location: Binary('Ux+yEeNACg==', 0)
reason: Binary('WBuqAeM=', 0)
```

**Figure 5.7** – Encrypted Data Store

## 5.3 PERFORMANCE ANALYSIS

This section highlights the system components used for the design of proposed cloud based Health Records Sharing system.

Here, the time complexity of various processes of medical encryption algorithms are analyzed and performance metrics are evaluated.

### 5.3.1 THEORETICAL ANALYSIS

Cryptographic algorithms are easy to compute in low memory usage.

On carrying out comparative analysis, the proposed approach is investigated on diverse parameters to evaluate the performance of various cryptographic algorithms and noted down the encryption and decryption modules runtime to estimate the efficiency of the computed algorithms in whole.

The computational algorithm analysis is carried out with different parameters and these modules are categorized to evaluate encryption and decryption modules. The parameters are:

- Development
- Key length
- Rounds
- Block Size
- Encryption Ratio
- Security Level
- Attacks Found

### 5.3.2 PRACTICAL ANALYSIS

The desired secure encryption algorithms were implemented in python and backend implementation is carried out with Flask Web Application. It allows for the user to interact with the proposed system and run the encryption algorithms against various use-cases and on analyzing the time taken to encrypt the text in the given text file will be carried out.

For the reason that the initializing process (Setup) is run only once for a certain IBE system, it is less valuable to test its performance so it is just ignored here. Therefore, by testing performance of Extract, Encrypt and Decrypt algorithms the desired Figure 5.8 is obtained. For each algorithm of a certain IBE scheme, the execution time under different parameters are recorded.

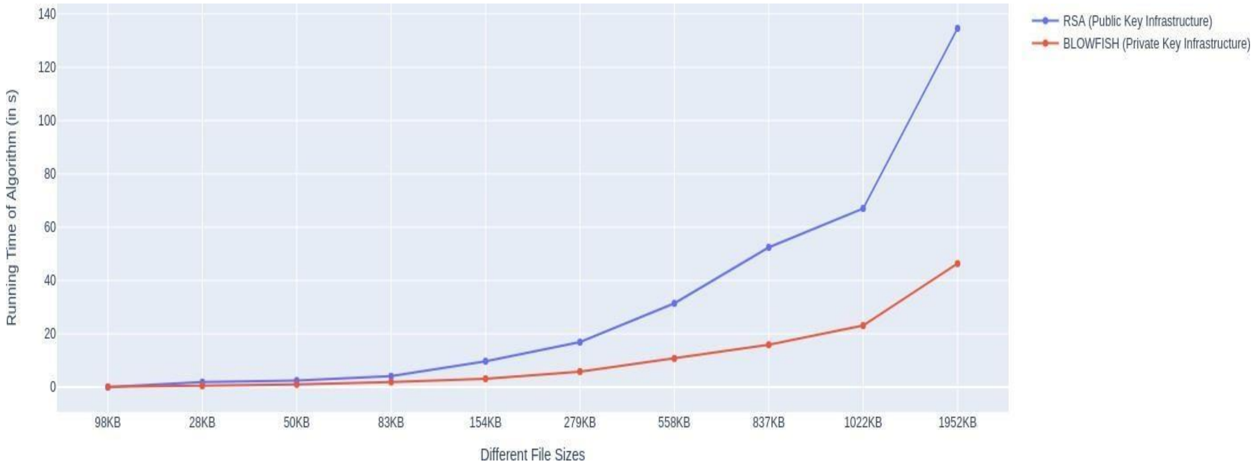


**Figure 5.8** – Running Time Comparison of Algorithms

Blowfish algorithm is observed to be the fastest algorithm.

The proposed algorithm was run against each files of different sizes and their execution times is noted down to derive graph. The throughput was calculated in MB/second and it is depicted in Figure 5.9,

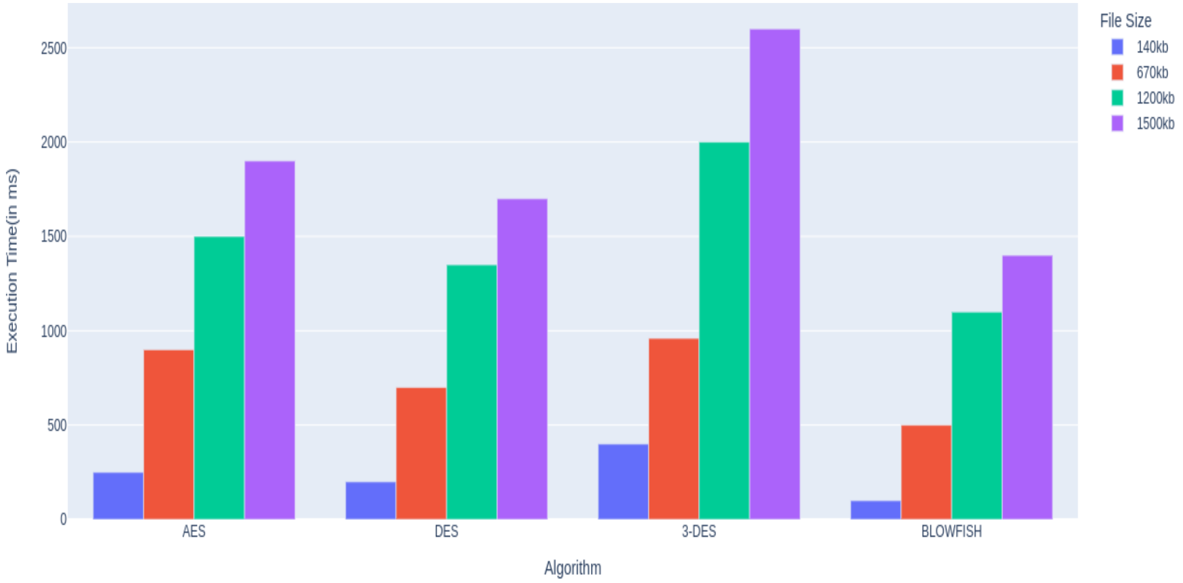


**Figure 5.9** – Encryption time for reports in pdf format calculated in ms

The algorithm's speed during encrypt and decrypt process is calculated of throughput.

**Throughput = Total Text files size (in MB) / Total Evaluation Time of Algorithm (in ms). (5.1)**

As the cryptographic technique's throughput value is increased, the power consumption of that technique will be decreased depending on the decreased time during encryption and decryption processes. On observing this calculation, the throughput of Blowfish algorithm is high when handled with voluminous amount of data.

### 5.3.3 ENVIRONMENT OF EXECUTION

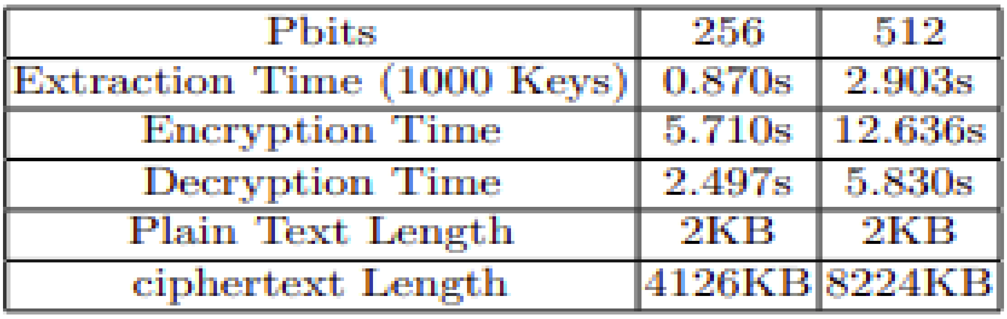
Processor - Pentium Dual T2330 @ 1.60GHz Memory - 3GB RAM

OS - Linux Interpreter - python

Timing - use command time, with accuracy of 1ms

File I/O work in memory, so execution time can be ignored

**Table 5.1** – TC-IBE Scheme Time Analysis



| Pbits | 256 | 512 |
|---|---|---|
| Extraction Time (1000 Keys) | 0.870s | 2.903s |
| Encryption Time | 5.710s | 12.636s |
| Decryption Time | 2.497s | 5.830s |
| Plain Text Length | 2KB | 2KB |
| ciphertext Length | 4126KB | 8224KB |

The above table depicts the running time of TC-IBE encryption and decryption methods involving plaintext of 2KB size to result in ciphertext.

**Table 5.2** – Modified BF Encryption Time

| Algorithm | Megabytes(2^20 bytes)Processed | Time Taken | MB/Second |
|---|---|---|---|
| Existing BF | 256 | 4.436 | 57.710 |
| Modified BF | 256 | 3.976 | 64.386 |

The above table depicts the execution time taken for modified Blowfish algorithm to execute in python run time environment, these algorithm execution parameters differ diversely on accordance with computational limits.

## CHAPTER 6

# CONCLUSION AND FUTURE WORK

### 6.1 CONCLUSION

The Secure channel for health records sharing among multiple hospital authorities and caretakers is facilitated with Timestamp based Cocks Identity based encryption system. This secure TC-IBE approach-based engaging and interactive web console will facilitate communication between patients and doctors to carry out diagnosis even in critical conditions. Also, the deployment of an interactive multi-user sharing system with secure encryption algorithms will ensure the authenticity of the objects involved in the network. It can be utilized in large-scale industries for sharing data in a wide range without the fear of tampering with data.

The identity-based approach allows the various entities to engage in the network to read/query the data. A secure sharing channel for health records is maintained to assist hospital authorities to ensure the integrity of data and effective drug prescription. Physicians will be provided with fast and more comprehensive access to patients' medical info in times of crisis. This in turn reduces the administrative burden. This system will rapidly improve the diagnosis performance of the healthcare facility in the rural areas as the primary health care centers in the rural areas can communicate with urban authorities regarding specific disease outcomes and precautions.

## 6.2 FUTURE WORK

The proposed system stores the already recorded health parameters in the secure cloud environment after proper encryption but it can be extended to record vital signs of patients with biosensors and store them in the cloud for continuous monitoring of patients in Covid isolation centers and remote rural areas.

The system can be structured further to arrive at Infographics and deploy it to the Cloud Platform. Also, better visualization charts to highlight the identified abnormalities in patients identified with defined threshold limits for health parameters can be added to the proposed architecture to serve those patients with utmost care.

The evaluation of the system can be further carried out with the engagement of multiple users and simulating the security attacks to arrive at the precise efficiency of the system. Enhancement of the system to detect abnormalities in the patient’s health using recorded vital signs and providing alerting mechanism to alert the related doctors, and caretakers to provide emergency healthcare services will be carried out. This will help to predict the onset of the most prevalent clinical events among the people of India and help the central health research authorities to find out insights about pandemics and outbreaks of disease in certain regions.

# REFERENCES

**1.** A. chouhan, A. kumari and M. Saiyad (2019), "Secure Multiparty Computation and Privacy Preserving scheme using Homomorphic Elliptic Curve Cryptography," *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pp. 776-780, doi: 10.1109/ICCS45141.2019.9065645.

**2.** Al-Issa, Yazan and Ottom, Mohammad Ashraf and Tamrawi, Ahmed (2019), “eHealth Cloud Security Challenges: A Survey” in Journal of Healthcare Engineering, vol. 2019, pp. 7516035, doi: 10.1155/2019/7516035.

**3.** Amiri R and O. Elkeelany (2021), "FPGA Design of Elliptic Curve Cryptosystem (ECC) for Isomorphic Transformation and EC ElGamal Encryption," in IEEE Embedded Systems Letters, vol. 13, no. 2, pp. 65-68, doi: 10.1109/LES.2020.3003978.

**4.** D. Boneh and M. Franklin (2003), “Identity-based encryption from the Weil pairing,” SIAM J. Computing., vol. 32, no. 3, pp. 586–615.

**5.** Deng H(2020), "Identity-Based Encryption Transformation for Flexible Sharing of Encrypted Data in Public Cloud," in IEEE Transactions on Information Forensics and Security, vol. 15, pp. 3168-3180, doi: 10.1109/TIFS.2020.2985532.

**6.** Ermatita, Ermatita & Prastyo, Yugo & Pradnyana, I & Adrezo, Muhammad (2020), Diffie-Hellman Algorithm for Securing Medical Record Data Encryption keys. 296-300. doi: 10.1109/ICIMCIS51567.2020.9354297.

**7.** J. Wei, W. Liu, and X. Hu (2018), “Secure data sharing in cloud computing using revocable-storage identity-based encryption,” IEEE Trans. Cloud

Computing., vol. 6, no. 4, pp. 1136–1148.

**8.** J. Lai, Y. Mu, F. Guo, W. Susilo, and R. Chen (2017), “Fully privacy-preserving and revocable ID-based broadcast encryption for data access control in smart city,” Pers. Ubiquitous Computing, vol. 21, no. 5, pp. 855–868.

**9.** Kashfia Sailunaz and Musaed Alhussein and Md. Shahiduzzaman and Farzana Anowar and Khondaker Abdullah Al Mamun (2016), “CMED: Cloud based medical system framework for rural health monitoring in developing countries”, in Journal of Computers & Electrical Engineering, vol. 53, pp: 469-481, doi: 10.1016/j.compeleceng.2016.02.005.

**10.** K. Li, W. Zhang, C. Yang, and N. Yu (2015), “Security analysis on One-to-Many order preserving encryption-based cloud data search,” IEEE Trans. Inf. Forensics Security, vol. 10, no. 9.

**11.** K. Lee (2020), "Comments on “Secure Data Sharing in Cloud Computing Using Revocable-Storage Identity-Based Encryption”," in IEEE Transactions on Cloud Computing, vol. 8, no. 4, pp. 1299-1300, doi: 10.1109/TCC.2020.2973623.

**12.** Mayank Kumar Kundalwal and Kakali Chatterjee and Ashish Singh (2019), “An improved privacy preservation technique in health-cloud”, in ICT Express, vol. 5, pp - 167-172, doi: 10.1016/j.icte.2018.10.002.

**13.** M. Keshk, B. Turnbull, E. Sitnikova, D. Vatsalan and N. Moustafa (2021), "Privacy-Preserving Schemes for Safeguarding Heterogeneous Data Sources in Cyber-Physical Systems," in IEEE Access, vol. 9, pp. 55077-55097, doi: 10.1109/ACCESS.2021.3069737.

**14.** Maiti S and Misra S (2020), "P2B: Privacy Preserving Identity-Based Broadcast Proxy Re-Encryption," in IEEE Transactions on Vehicular Technology, vol. 69, no. 5, pp. 5610-5617, doi: 10.1109/TVT.2020.2982422.

**15.** Paragas, Jessie. (2020), An Enhanced Cryptographic Algorithm in Securing Healthcare Medical Records. 1-6, doi: 10.1109/ICVEE50212.2020.9243228.

**16.** P. Xu, T. Jiao, Q. Wu, W. Wang, and H. Jin (2016), “Conditional identity-based broadcast proxy re-encryption and its application to cloud email,” IEEE Trans. Computing., vol. 65, no. 1, pp. 66–79.

**17.** Thangapandiyan M, P. M. R. Anand and K. S. Sankaran (2018), "Enhanced Cloud Security Implementation Using Modified ECC Algorithm," 2018 International Conference on Communication and Signal Processing (ICCSP), pp. 1019-1022, doi: 10.1109/ICCSP.2018.8524212.

**18.** Ullah, Syed Sajid and Ullah, Insaf and Khattak, Hizbullah and Khan, Muhammad Asghar and Adnan, Muhammad and Hussain, Saddam and Amin, Noor Ul and Khattak, Muazzam A. Khan (2020), "A Lightweight Identity-Based Signature Scheme for Mitigation of Content Poisoning Attack in Named Data Networking With Internet of Things," in IEEE Access, vol. 8, pp. 98910-98928, doi: 10.1109/ACCESS.2020.2995080.

**19.** Zhang, R. Xue, and L. Liu (2018), “Searchable encryption for healthcare clouds: A survey,” IEEE Trans. Services Computing., vol. 11, no. 6, pp. 978–996.

**20.** Zhang L and Zhang Z (2017), "Security Analysis of an ID-Based Two-Server Password-Authenticated Key Exchange," in IEEE Communications Letters, vol. 21, no. 2, pp. 302-305, doi: 10.1109/LCOMM.2016.2594789.