S-DIHE: Secure Deduplication of images based on Homomorphic Encryption

by

Nimmi Patel
202011016

A Thesis Submitted in Partial Fulfilment of the Requirements for the Degree of

MASTER OF TECHNOLOGY
in
INFORMATION AND COMMUNICATION TECHNOLOGY
to

Dhirubhai Ambani Institute of Information and Communication Technology

May 2022

Declaration

I hereby declare that

i) the thesis comprises of my original work towards the degree of Master of Technology in Information and Communication Technology at Dhirubhai Ambani Institute of Information and Communication Technology and has not been submitted elsewhere for a degree,

ii) due acknowledgment has been made in the text to all the reference material used.

NPatel

Nimmi Patel

Certificate

This is to certify that the thesis work entitled S-DIHE: Secure Deduplication of images based on Homomorphic Encryption has been carried out by Nimmi Patel for the degree of Master of Technology in Information and Communication Technology at *Dhirubhai Ambani Institute of Information and Communication Technology* under my/our supervision.

Priyanka Singh

Prof. Priyanka Singh
Thesis Supervisor

Acknowledgments

Remarkably, the M.Tech Thesis was a challenging and rewarding task. As a result, I would like to extend my sincere gratitude to all those who helped in different roles with the thesis work and designing new promising methodologies.

To begin with, I am grateful to Prof. Priyanka Singh for guiding and mentoring me throughout the work by providing a comprehensive direction and engaging me in discovering different possibilities till the end. Furthermore, I would like to thank my fellow friend Jash Rathi for advising me to implement better methodologies. Also, I would like to thank my family, professor, and friends for giving their time to help me become a better person. Finally, I am grateful to thank the Dhirubhai Ambani Institute of Information and Communication Technology for giving me this opportunity.

Contents

Abstract

The cloud space is occupied with multiple versions of the same data, including images, text files, et. Cetera. Though cloud service providers (CSPs) are providing very cost-effective storage space but combating the unprecedented growth of data requires ways to optimize the data and minimize storage costs. Therefore the demand for data deduplication arises, which enables the removal of extra copies of the same data and manages the storage space efficiently. In this thesis, we proposed two frameworks, S-DIHE and Privacy-Preserving Disease Prediction and Secure Data Deduplication of Health Data. These frameworks deal with secure data deduplication and maintaining the privacy of users' data. In the first framework, we propose secure data deduplication approach to identify the duplicate or near-duplicate images without actually looking at the underlying content. It is based on the client-server model where the client encrypts the image prior to outsourcing it to the server, and the server maintains only the unique copies at its end. The client's privacy is preserved in the entire process as the data is encrypted using homomorphic encryption. Further, the data integrity is checked both at the server-side while uploading and client-side during download. Experiments are carried out to verify the proposed approach's performance against a variety of potential attack scenarios, such as poison attacks, dictionary attacks, side-channel attacks, and frequency analysis attacks. In the second framework, we propose a framework that predicts the diseases based on the user's symptoms without compromising the user's privacy. It also performs secure data deduplication on the disease prescription to minimize the storage requirement. Instead of giving prescriptions to every patient individually, only a unique copy of the prescription is maintained at the CSP, and the access table is updated accordingly for the patients. The efficiency of the proposed framework is validated through the experiments.

List of Tables

List of Figures

CHAPTER 1
Introduction

Cloud computing has enabled several services to be delivered to numerous enterprises and organizations at an affordable price. They can avail the computational resources, have the access rights to several softwares installed on the high-end remote machines, and above all enjoy more storage space at their discretion. But no matter how much storage space is made available, it is never sufficient as the data is growing at an unprecedented scale. Hence, data deduplication is urgently needed that can help maintain only a single copy of the floating data on these cloud servers and optimize the usage of the storage space. For instance, if two images have slight differences in some blocks, as shown in Fig.1.1, it is termed a near-duplicate image and hence it is considered for deduplication.

File-level deduplication and block-level deduplication are two types of data deduplication strategies. In file-level deduplication, only a single copy of each file is saved on the cloud servers, whereas in block-level deduplication, each file is broken down into blocks to perform fine-grained deduplication where the size of the blocks might be fixed or variable. Several schemes have been proposed in the literature to address data deduplication [1], [2].Though deduplication provides the advantage of optimizing storage space at the CSP, still it poses a serious threat to the outsourced data. As the data is not encrypted, it is at the discretion of the CSPs, giving every reason to the privacy-aware society not to adopt such services [3]. Hence, secure data deduplication is in much demand where storage space can be saved along with securing the outsourced data. Secure data deduplication works on top of encrypted data, so even the CSP can't see the underlying content [4], [5], [6]. However, there is a big challenge working with encrypted data. Most of the encryption schemes are based on randomization that leads to different ciphertexts in different instances of encryption. This leads to the identification of the same data very challenging and hence performs deduplication. As a solution, convergent encryption (CE) was introduced where the secret key was obtained based on the content of the data and hence, it would lead to the same ciphertext at any iteration of encryption. HCE1, HCE2 and randomized convergent encryption (RCE) are some variants of CE, but they can't handle poison attack [7]. Also, the integrity check of data while uploading and downloading was present only at one end, either client-side or server-side. Agarwala et al. addressed the poison attack and dual integrity check problem, but it could not handle attack scenarios such as frequency analysis, side-channel and dictionary attacks [8]. Comparison of protocols based on different aspect is shown in Table 1.1.

Table 1.1: Comparison of protocols.

	CE	HCE1	HCE2	RCE	DICE	**S-DIHE**
Data integrity check (Client)	No	No	Yes	Yes	Yes	**Yes**
Data integrity check (Server)	Yes	No	No	No	Yes	**Yes**
Poison attack	Yes	No	No	No	Yes	**Yes**
Frequency analysis attack	No	No	No	No	No	**Yes**
Side channel attack	No	No	No	No	No	**Yes**
Dictionary attack	No	No	No	No	No	**Yes**

Digital healthcare is an essential source in today's healthcare system. It is a novel concept of health informatics that refers to the technologies to deliver health services and information to patients. More benefits of adopting the digital healthcare system include online services for teleconsultation, e-prescription, e-referral, remote monitoring, telecare, etc. Moreover, the digital healthcare system delivers a high level of security for the patient records. The use of a modern healthcare system increases the possibility of fulfilling the demands and requirements of both the patients and healthcare providers [9], [10].

In this thesis, we proposed two different frameworks for secure data deduplication.

- S-DIHE: Secure Deduplication of images based on Homomorphic Encryption
- Privacy-Preserving Disease Prediction and Secure Data Deduplication of Health Data

In S-DIHE, we can perform secure data deduplication even if the ciphertext keeps on changing with every iteration of encryption. It is based on a robust homomorphic image hash proposed by Singh et al. [11]. The stability of the hash against some common manipulations such as scaling, corruption by noise, contrast changes and jpeg compression is also exploited to achieve better data deduplication.

Privacy-Preserving Disease Prediction and Secure Data Deduplication of Health Data is a privacy-preserving framework for disease prediction based on clinical symptoms of the user. Also, as maintenance of health records incurs a huge storage cost, the framework addresses this issue by performing secure data deduplication on the disease prescription maintained in the database. The proposed framework consists mainly of four entities: patient, medical staff, medical officer and the CSP. The role of the medical staff is to collect the patient's details and clinical symptoms. It encrypts patients' details and sends it to the medical officer. Based on the collected symptoms, the medical officer generates a symptoms string and sends it to the CSP. The CSP detects the disease based on the symptoms string.

The following are the major contributions of the thesis:

1. **Block-level secure data deduplication:** We propose a secure block-level data deduplication scheme. It divides the encrypted image into blocks on the server and performs deduplication based on extracting hash values from these blocks without seeing the underlying content.

2. **Secure against attacks:** The proposed scheme is secure against potential attacks scenarios, including poison attacks, dictionary attacks, side-channel attacks, and frequency analysis attacks.

3. **Dual side data integrity check:** The proposed scheme supports data integrity check at both ends, the client as well as the server. During the upload phase, it is checked at the server-side and for the download phase, it is validated at the client-side.

4. **Robust against common manipulations:** As the proposed scheme is based on robust image hashing, it helps to achieve better data deduplication. Duplicates are found based on the hash value that remains stable to common manipulations such as scaling, corruption by noise, contrast changes and jpeg compression.

5. **Ability to detect disease:** We propose a framework to predict diseases based on the clinical symptoms collected from the user. For every user, a weighted sum of the symptoms is computed by the medical officer and compared with the pre-defined threshold values at the CSP to predict the diseases.

6. **User privacy:** The proposed framework maintains the user's privacy throughout the process of disease prediction. As the data moves up the channel from the medical staff to the medical officer and CSP, it is encrypted so that no data leakage happens. All the computations are done on top of the encrypted data and no unauthorized person gets access to the underlying content. Thus, preserving the user's privacy.

7. **Secure data deduplication:** In the proposed framework to optimize the storage requirements of the health records maintained by the medical officer, secure data deduplication is performed on the disease prescription. Instead of prescribing for every patient individually, only the access table of the prescription is updated in case of a user diagnosed with the same disease.

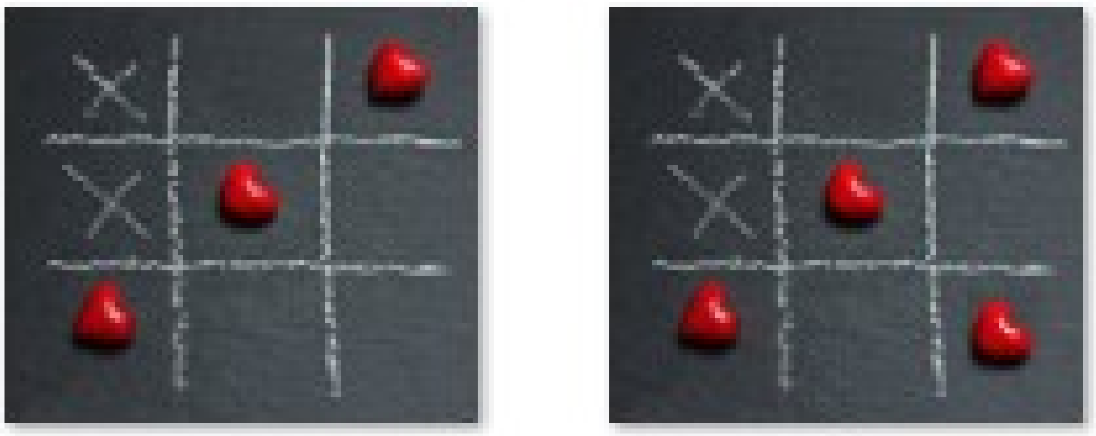


Figure 1.1: Two near-duplicate images with slight changes in some blocks

The rest of the thesis is written as follows: The chapter 2 discusses the preliminaries of the attacks and paillier encryption. In chapter 3, related work is described. For chapters 4, 5, and 6, we present the proposed frameworks, results and security analysis respectively. Finally, chapter 7 brings the thesis to a close with the conclusion.

CHAPTER 2
Preliminaries

This chapter discusses about attack scenarios and paillier encryption.

2.1 Attack Scenarios

- **Poison attack:** It is a combination of erasure attack and duplicate faking attack.

  Erasure attack: When the user sends the data to the CSP, an attacker can remove some of the data when the data is on the insecure communication channel.

  Duplicate faking attack: An attacker can inject malicious inputs to the data that modifies the actual data. The client fails to recognise original data that leads inaccurate results.

- **Frequency analysis attack**: When the user encrypts the same data multiple times than the generated ciphertext will also remains same. On receving the same ciphertext multiple times the attacker can map between the original data and its ciphertext.

- **Side channel attack:** An attacker tries to gain informatin of user's cruical data. The two main reason for this attack is: 1) data that is transferred from the user to the CSP over the network is visible to everyone 2) A tag value is used to check the existance of data on the CSP. An adversary can check the existance of data by trying different values of tag. An attacker computes duplicate tag T^* which he sends to the CSP. If T matches with T^* then the CSP reject the request as data is already present and provides metadata of the file to an adversary. As a result, an adversary knows the existance of data as well as the partial content of the data.

- **Dictionary attack:** To begin with the upload protocol, the user must send the tag value to the server. When the server receives tag value it perfom search operation in his maintaned tag table. If the tag value matches any other tag value stored on the server, the client is updated with all the data information and makes the user an authorised user. The disadvantage of this approach is that authorization to data is based on tag value, and the computed tag value is simply a hash output. By guessing the tag value and running the protocol several times, any malicious user can take advantage of this. This seriously compromises the protocol's security. To achieve data ownership under the proposed scheme S-DNDI, the tag value should get match with any of the stored tag value at the CSP and then the user must prove its authenticity by replying to the CSP challenge.

2.2 Paillier Encryption

The Paillier cryptosystem is an additive partial homomorphic encryption scheme. It consists of three main phases:

1. Generation of public-private key pair.
2. Encryption of a message.
3. Decryption of a message.

First two large prime numbers p and q are selected, then public and private key are evaluated. Public key n and g are calculated as follows:

$$n = p \times q \tag{2.1}$$

$$g = n + 1 \tag{2.2}$$

The private keys λ and μ are computed as follows:

$$\lambda = lcm(p-1, q-1) \tag{2.3}$$

$$\mu = mod(\alpha, n) \tag{2.4}$$

Here, *lcm* implies the least common multiple and *mod* denotes the modulo operator.

With the public (n, g) and private (λ, μ) keys generated, now we describe how to encrypt and decrypt a message.

Encryption:
A plaintext message m can be encrypted as follows:

1. Let m be a message to be encrypted where $0 \leq m \leq n$.
2. Select random r where $0 < r < n$
3. Compute cipher text as $C = g^m r^n mod\, n^2$.

Decryption:
A ciphertext message C is decrypted as follows:

1. Compute plain text message as:

$$I = L(c^{\lambda}\ modn^2) \cdot \mu\ mod\ n \tag{2.5}$$

CHAPTER 3

Related Work

As the data keeps growing rapidly, data deduplication is much in demand. Data deduplication provides the advantage of optimizing the storage space at the cloud server. But still it becomes a serious threat to outsourcing the data from the user end as the data is not encrypted. As a result, demand for secure data deduplication was made, where data security is preserved along with saving storage space [12], [13]. To work on encrypted data is difficult as most of the encryption schemes are based on randomization, which leads to different ciphertext. As a result, it will make it hard to perform data deduplication.

To address this problem, several convergent encryption (CE) scheme was introduced [14],[15]. Several message locked encryption (MLE) based data deduplication schemes have been proposed in the literature [16], [17]. Keelveedhi et al. [18] introduced the MLE technique, where the key used for encryption and decryption was derived from the message itself. But these schemes are prone to poison attacks where an adversary can manipulate the stored file or delete it permanently. In both cases, the authentic user ends up losing their original files.

Traditional encryption algorithms discourage data deduplication as they generate different ciphertexts even for the same content. So kanakamedala et al. [19], proposed paper that aims at an attribute-based storage system with secure deduplication in a hybrid cloud scenario, where a private cloud takes care of duplicate detection, and a public cloud takes care of storage.

Data deduplication is carried out in two ways, either it can be on the server side or the client side. On the client side, computation is done by the client where the client computes tags which are nothing but the hash values. It is then sent to the server, where based on the tag, deduplication is performed. On the server side, the clients send the data to the server, and then the server performs deduplication by keeping unique copies of the data. Agarwala et al. [20] proposed a DICE protocol where the computations are performed on the client side. They performed deduplication at the file level by applying the hash function on the

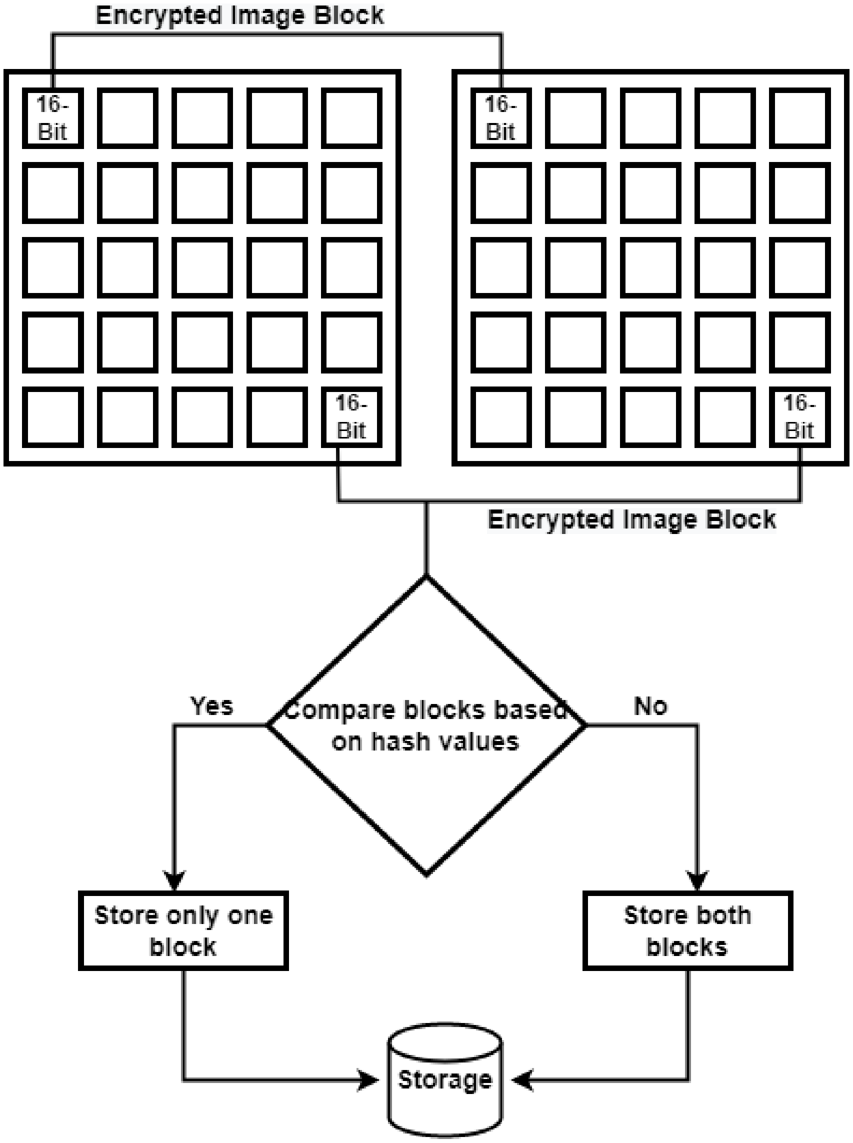


Figure 3.1: Deduplication of two images at the block level.

whole file. After checking at the server side whether hash values are the same. Applying the hash function to the entire image will not be appropriate in this scenario since the hash value may change even if the two images are different in pixel value. Deduplication will be unsuccessful. To solve this problem, the image is fragmented into blocks, and deduplication is done by comparing the blocks that has same hash values.

Coming to secure data deduplication on images, several research work has been proposed. For instance, Gang et al. [21] to perform image deduplication, the CE technique was used with Attribute-Based Encryption. In another work, Chen et al. [22] they introduce a hashing and clustering-based secure image deduplication scheme. They merge an unique perceptual hash algorithm based on the Local Binary Pattern in this study. Where the image is encrypted, the hash value of the image is used as the fingerprint for deduplication. Images are clustered to reduce the time complexity of deduplication. The proposed scheme can improve deduplication accuracy while also securing images.

Manikyam et al. [23] image Decompression Model with Reversible Pixel Interchange Decryption model using data deduplication (IDRPID-DD) during the initial stage, the original image is segmented into blocks of the same size, and sub-classification is used to extract accurate images within the constraints. A random matrix is used to swap the pixels of neighbouring subblocks. Following that, using a random matrix, each pixel is randomly exchanged for nearby blocks, and each block is encrypted using the provided function that provides security during storage and data transmission.

Ullah et al. described a symmetric key-based encryption method for transmitting accumulated data in the context of healthcare sensing [24]. They proposed a Controlled Window Size based Chunking (CWSC) technique which decreases duplicate values. It uses a min and max delta value on window sizes to make sure that chunks are of decent size. A window size larger than 75% of the delta value is accepted. If the window size is smaller, the next cut point is determined using a reference value, such as a similar temperature in the next patient's data. When the chunk size exceeds the delta value, the data sequence is backtracked until it reaches a nearby delimiter. Attigeri et al. focused on classifying Electronic Medical Record (EMR) based on specialisation using the k-nearest neighbor (KNN) algorithm [25]. Thereafter, they used deduplication to optimize storage and exploited DNA encryption algorithm to protect the data prior to transferring to Hadoop.

A large number of databases can be stored at the cloud server and recovered on a daily basis. But with health data, this task becomes quite challenging due to the privacy and security concerns [9]. Kim et al. used attribute-based encryption to address the privacy violations and implemented a novel deduplication method to tackle the storage requirements [26]. As the user's map structure and files are not saved on the server, the file uploader list cannot be extracted via the server's meta-information analysis, protecting the privacy of the users. Furthermore, to fix the issue of file ownership, they make use of a personal identification number (PIN). Meingast et al. looked at the security and privacy concerns associated with Electronic Health Records (EHR's), patient portals and wireless detecting systems for patient monitoring [27].

Hong et al. used blockchain technology to propose health data based privacy-preserving framework. They encrypted the health data, which allows restricted controlled access. Users can automatically add or remove authorized doctors by enabling user transactions for key management. Moreover, by implementing a health chain, no one can modify the IoT data and doctor diagnosis to prevent medical conflicts [28]. Hass et al. also proposed a privacy-preserving information system for restricted access of personal data to the third parties [29].

In this thesis, we have proposed a server-side secure block-level deduplication method that eliminates near-duplicate images in encrypted form while maintaining image security. It's also resistant to attacks even if the same images are encrypted more than once because ciphertexts change each time which makes it more difficult for an attacker to harm or duplicate the data. We used robust homomorphic image hashing as the stability of hash is strong against resizing the image pixel, apply colour contrast or changing the image format. It will not reflect in the hash values of the images. We have used a paillier algorithm to encrypt each block.

We also proposed Privacy-Preserving Disease Prediction and Secure Data Deduplication of Health Data in which the disease prediction is done based on the symptoms collected from the patient by the medical staff. Prior to transferring the patient's personal details up the channel to medical officer and CSP, the private attributes like id, age and mobile number are encrypted using homomorphic encryption. In the entire disease prediction protocol, the user privacy is fully preserved. Also, secure data depulication is done while prescribing for the disease, so that the storage requirements are also optimized and efficiency is maintained.

CHAPTER 4

Proposed Frameworks

This chapter discusses the proposed framework S-DIHE and Privacy-Preserving Disease Prediction and Secure Data Deduplication of Health Data. It also discusses the entities involved and their specific roles.

4.1 S-DIHE: Secure Deduplication of images based on Homomorphic Encryption

4.1.1 Threat model

This subsection describes the involved entities.

User: The user is an entity who has access to the cloud services to upload the encrypted images. It is considered as an honest entity.

CSP: The CSP is an entity that provides cloud storage services to the user. It contains a database with user IDs and encrypted images hash values. The CSP must ensure that the users data should be secure from unauthorized users. The CSP is considered as a semi-honest entity. It runs the protocol honestly but is curious to gain the information.

Third party server: It receives the information about the image being near-duplicate or not. It employs the received information towards various applications such as maintaining only the unique images. It is also considered as semi-honest entity.

4.1.2 The Proposed S-DIHE

This subsection describes the procedure of the proposed framework.
User:

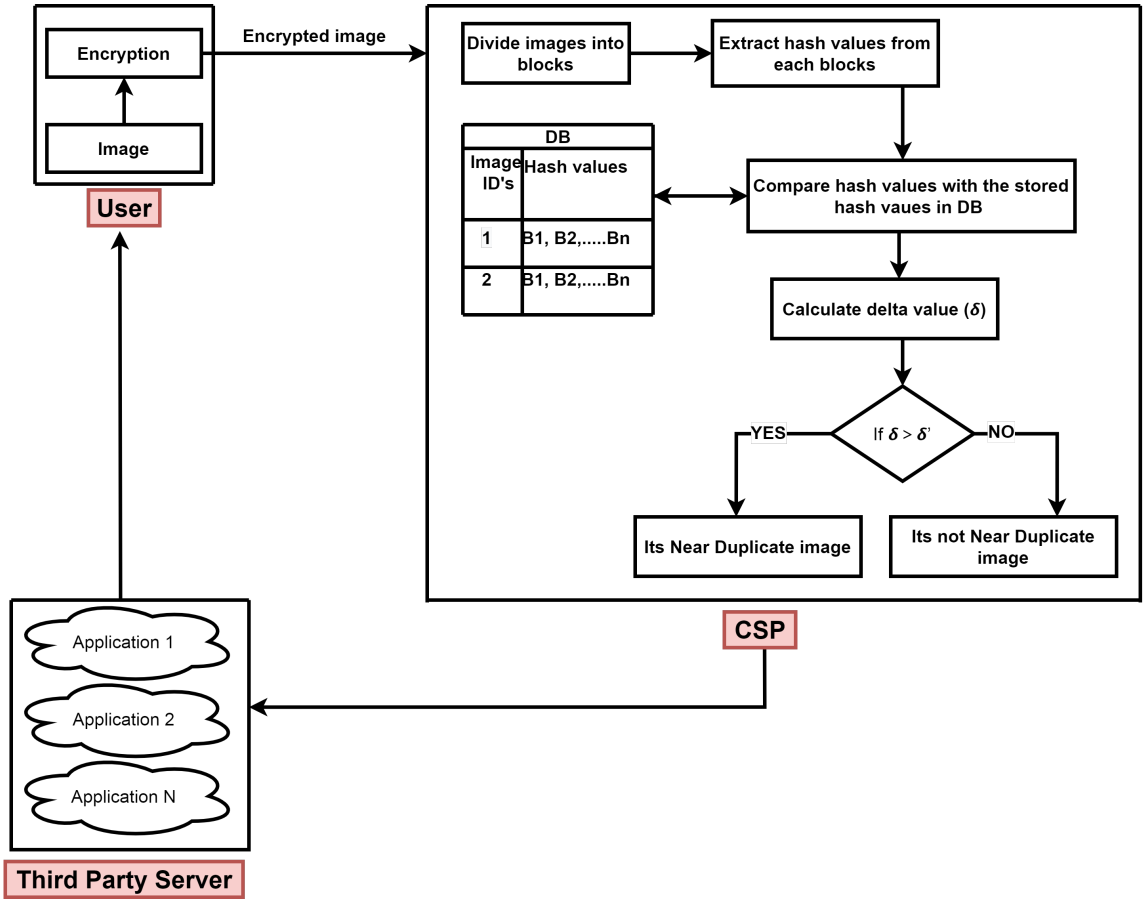


Figure 4.1: S-DIHE: System Architecture

Step 1: The user encrypts the image I.

$$C \leftarrow E(I) \tag{4.1}$$

where C is generated cipher image and E is the encryption function.

CSP:

Step 2: The CSP will divide the encrypted image C into n-blocks

$$S = B_1, B_2, ..., B_n \tag{4.2}$$

where B_i represents the i^{th} block of an encrypted image.

Step 3: The CSP will extract the hash values from each of the blocks.

$$X_i \leftarrow H(B_i) \tag{4.3}$$

where $H()$ is a hash function and X_i represent the i^{th} hash value.

Step 4: The CSP will compare hash value X_i with the stored hash value X_i' in database DB and compute the δ-Value.

Step 5: The δ-value is calculated out as follows:

$$\delta = \frac{No.\,of\ blocks\ with\ the\ same\ hash\ values}{Total\ no.\,of\ blocks} \tag{4.4}$$

Step 6: Based on the threshold, the CSP will decide whether the image is near duplicate image or not.

$$\forall\, \delta_i = \begin{cases} if\ \delta_i > threshold, & \text{its near duplicate image.} \\ else, & \text{not near duplicate image.} \end{cases} \tag{4.5}$$

Third party server:
Step 7: Third party server employs the information received from the CSP towards different applications for instance, performing secure data deduplication.

4.1.3 Proposed protocol for uploading and downloading images

The data integrity is checked on both ends: the CSP side while upload and the user side during download. The upload and download protocols are discussed below.

Upload:
In the upload protocol as shown in Fig.4.2, the user sends the encrypted image to the CSP if its not already at the CSP.
The image I is encrypted to ciphertext C using E encryption strategy as $C \leftarrow E(K, I)$, where K is the secret key from the user. Further, it computes tag T as the hash of the ciphertext as $T \leftarrow H(C)$ and sends this tag value to the CSP. The CSP, on receiving the tag value, searches for a tag in its maintained tag table. If the match is found, it simply updates the user with a data pointer; else, it will request the user to send a ciphertext value. On receiving the ciphertext value, the CSP will compute the tag as $T' \leftarrow H(C)$ and store that computed tag.

Download:

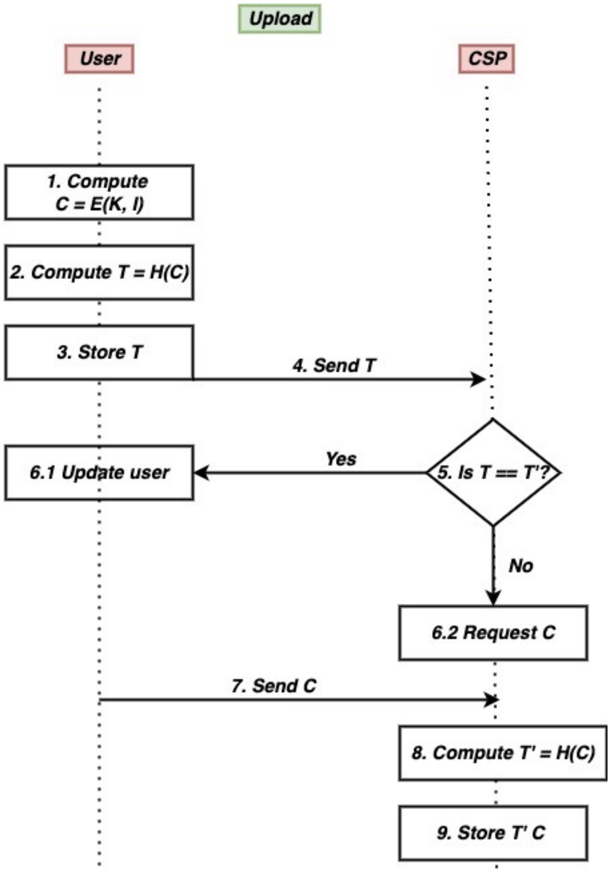


Figure 4.2: Upload - proposed protocol

In this phase as shown in Fig.4.3, the user attempts to download his data from the CSP.

The user sends its user ID and stored tag value to the CSP. The CSP performs a search operation for the tag in its maintained tag table; if the match is found, it generates the challenge $\pi c'$ (last block of cipher image C) and asks the user to respond back with πc (last block of cipher image C). If both the block values match, the CSP will provide a data pointer to the user; else, it will not provide it. If $T == T^{'}$, Then the user checks the data integrity by recomputing the tag $T^{''}$ as $T^{''} \leftarrow H(C)$. Here if $T == T^{''}$, then image I is retrieved else image is considered as corrupt.

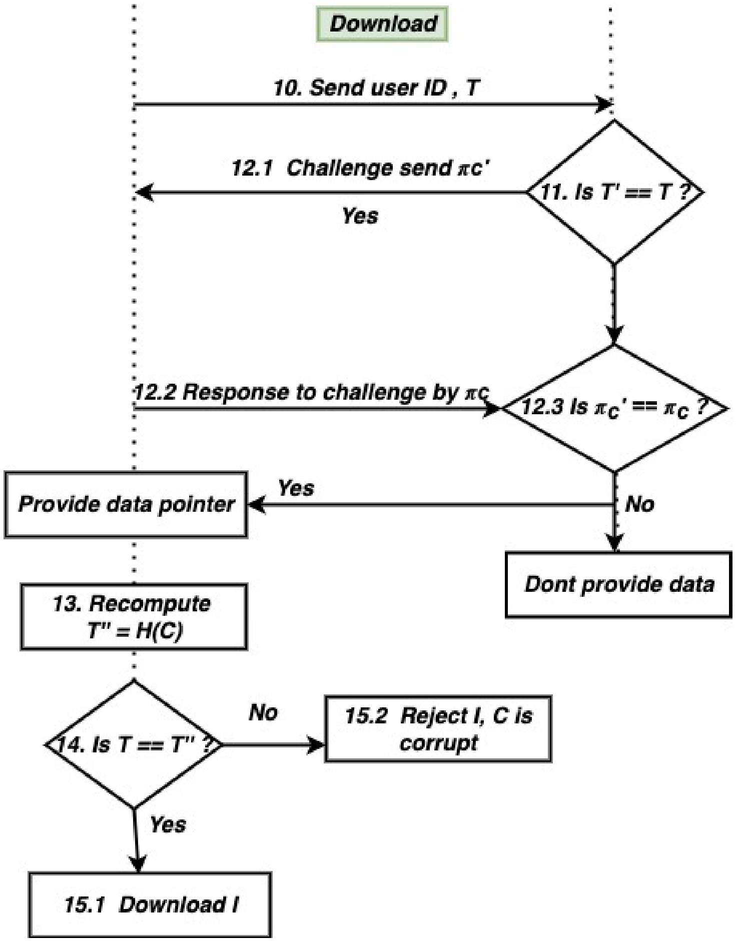


Figure 4.3: Download - proposed protocol

4.2 Privacy-Preserving Disease Prediction and Secure Data Deduplication of Health Data

4.2.1 Threat Model

This subsection describes the involved entities.
Patient:

- The patient is the entity who wants to line up for the medical checkup. He provides his details and clinical symptoms to the medical staff and expects a prescription if he gets diagnosed with a disease. The patient is considered an honest entity.

Medical Staff:

- The medical staff collects the patient's personal details and clinical symptoms. He encrypts the patient's details, i.e., id, age, and mobile no. and clinical symptoms will remain as it is. The medical staff is considered an honest entity.

Medical Officer:

- Based on the symptoms received from the medical staff, the medical officer generates a symptoms string and calculates the δ-value. Finally, he sends the patient's ID, δ-value, and symptoms string to the CSP. A medical officer is considered a semi-honest entity. He categorizes the patients based on their age into one of these age groups: 0-20, 21-40, 41-60, 60 & above. This is done to prioritize the doctor's appointment based on the severity of the disease and the age group. This is done on top of encrypted data received from the medical staff, ensuring the user's privacy.

CSP:

- The CSP is an entity that predicts the diseases based on symptoms string sent by the medical officer. It maintains a disease database that contains diseases and their corresponding strings. The received symptoms string is checked against this disease database. The CSP is considered a semi-honest entity.

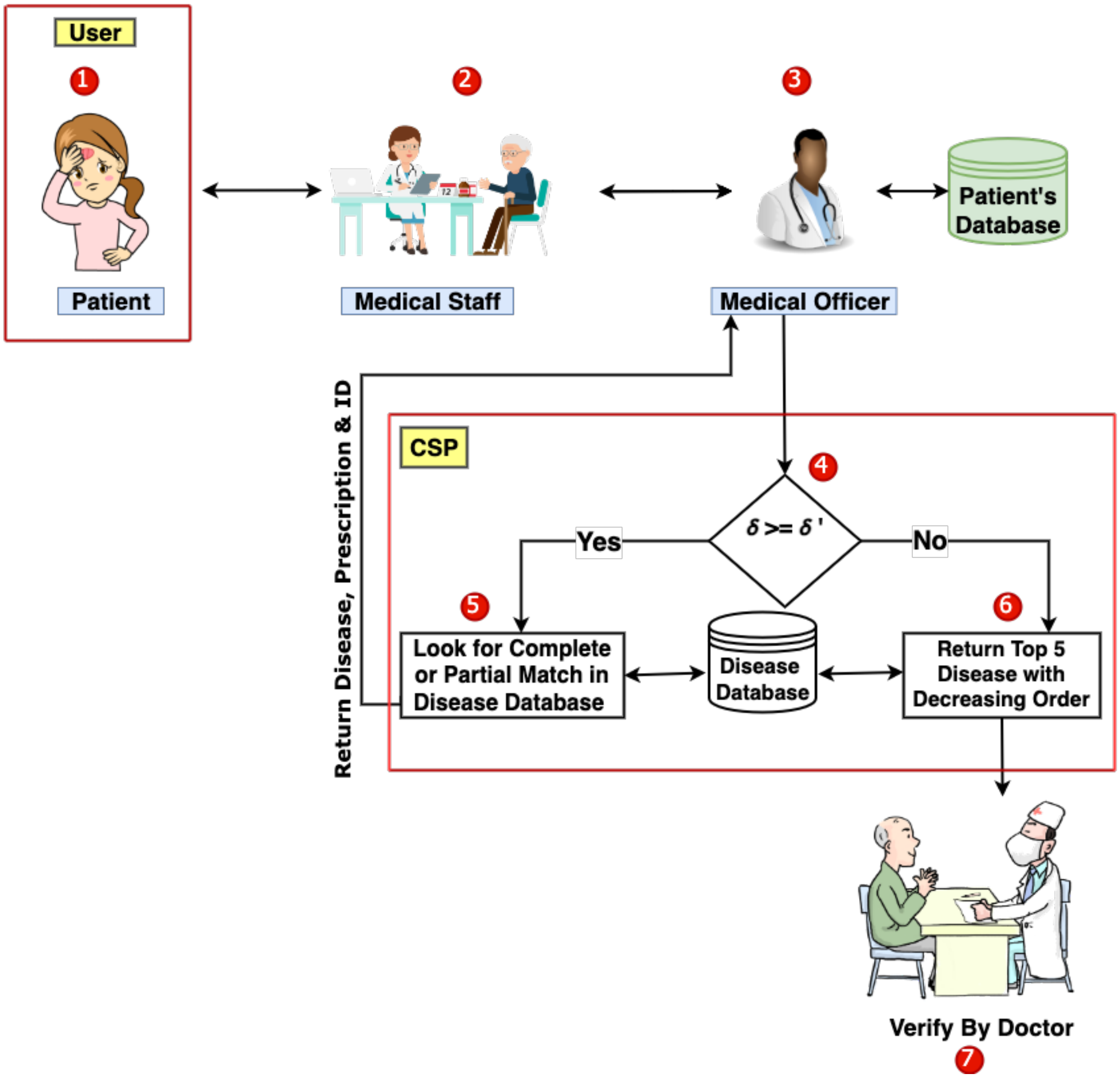

User
1
Patient
2
Medical Staff
3
Medical Officer
Patient's Database
Return Disease, Prescription & ID
CSP
4
δ >= δ '
Yes
No
5
Look for Complete or Partial Match in Disease Database
Disease Database
6
Return Top 5 Disease with Decreasing Order
Verify By Doctor
7

Figure 4.4: A overview of the proposed framework

4.2.2 Steps of the proposed framework

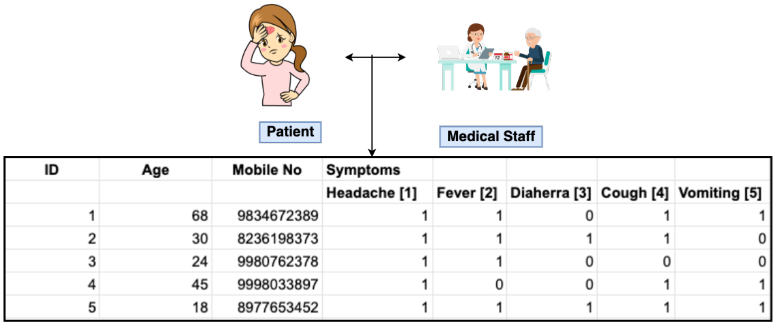
This subsection describes our proposed method.

Patient:

Step 1: The patient goes to the hospital for a medical checkup. He provides his necessary details i.e., id, age and mobile number, and his clinical symptoms as shown in Fig. 4.5.

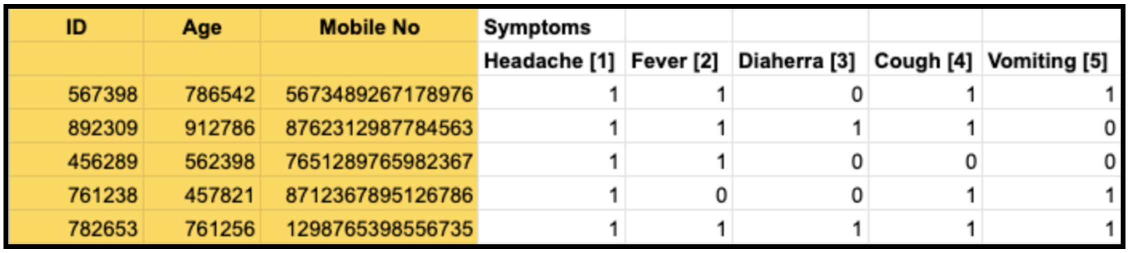
Medical Staff:

Step 2: After collecting patient's details, the medical staff encrypts his details, i.e., id, age and mobile number, as shown in Fig. 4.6. Thereafter, he sends the encrypted details of the patient along with his clinical symptoms to the medical



ID	Age	Mobile No	Symptoms				
			Headache [1]	Fever [2]	Diaherra [3]	Cough [4]	Vomiting [5]
1	68	9834672389	1	1	0	1	1
2	30	8236198373	1	1	1	1	0
3	24	9980762378	1	1	0	0	0
4	45	9998033897	1	0	0	1	1
5	18	8977653452	1	1	1	1	1

Figure 4.5: Patients providing their details and clinical symptoms to medical staff officer.

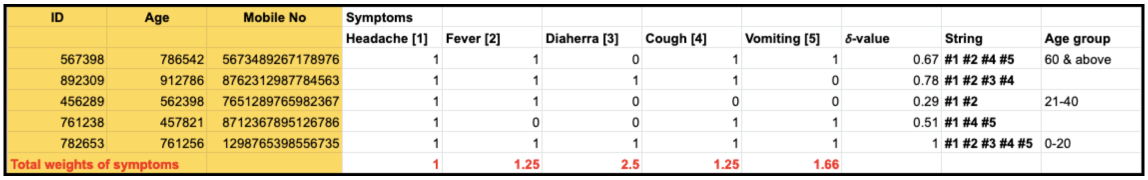


ID	Age	Mobile No	Symptoms				
			Headache [1]	Fever [2]	Diaherra [3]	Cough [4]	Vomiting [5]
567398	786542	5673489267178976	1	1	0	1	1
892309	912786	8762312987784563	1	1	1	1	0
456289	562398	7651289765982367	1	1	0	0	0
761238	457821	8712367895126786	1	0	0	1	1
782653	761256	1298765398556735	1	1	1	1	1

Figure 4.6: Encrypted patient details by the medical staff

Medical Officer:

Step 3: On receiving the patient's details from the medical staff, the medical officer will calculate the δ-value. After that he will generate a symptoms string based on the collected symptoms as shown in Fig. 4.7. Then he will send the symptoms string, δ-value and patient's ID to the CSP.



ID	Age	Mobile No	Symptoms							
			Headache [1]	Fever [2]	Diaherra [3]	Cough [4]	Vomiting [5]	δ-value	String	Age group
567398	786542	5673489267178976	1	1	0	1	1	0.67	**#1 #2 #4 #5**	60 & above
892309	912786	8762312987784563	1	1	1	1	0	0.78	**#1 #2 #3 #4**	
456289	562398	7651289765982367	1	1	0	0	0	0.29	**#1 #2**	21-40
761238	457821	8712367895126786	1	0	0	1	1	0.51	**#1 #4 #5**	
782653	761256	1298765398556735	1	1	1	1	1	1	**#1 #2 #3 #4 #5**	0-20
Total weights of symptoms			**1**	**1.25**	**2.5**	**1.25**	**1.66**			

Figure 4.7: Calculating δ-value and generating symptoms string

For instance, as shown in Fig. 4.7, δ-value and symptoms string of encrypted id-567398 is calculated as follows:

$$
\begin{aligned}
\delta =& \frac{Symptoms\ Detected}{Total\ Symptoms}\\
=& \frac{wt(Headache) + wt(Fever) + wt(Cough) + wt(Vomiting)}{Total\ weight\ of\ symptoms}\\
=& 5.16/7.6\\
=& 0.67
\end{aligned}
\tag{4.6}
$$

Here, weight(wt) of symptoms is calculated as follows:

$$
Weight\ \ of\ symptoms = \log_{10}(\frac{No.\ of\ diseases}{No.\ of\ diseases\ containing\ same\ symptoms}) \tag{4.7}
$$

Symptoms string is generated as follows:
The detected symptoms are marked as ’1’, and their corresponding index is concatenated to the symptoms string. For instance, in id-567398 the symptoms which are marked as ’1’ are #1 #2 #4 #5 as shown in Fig. 4.7.

CSP:

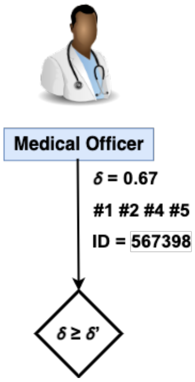
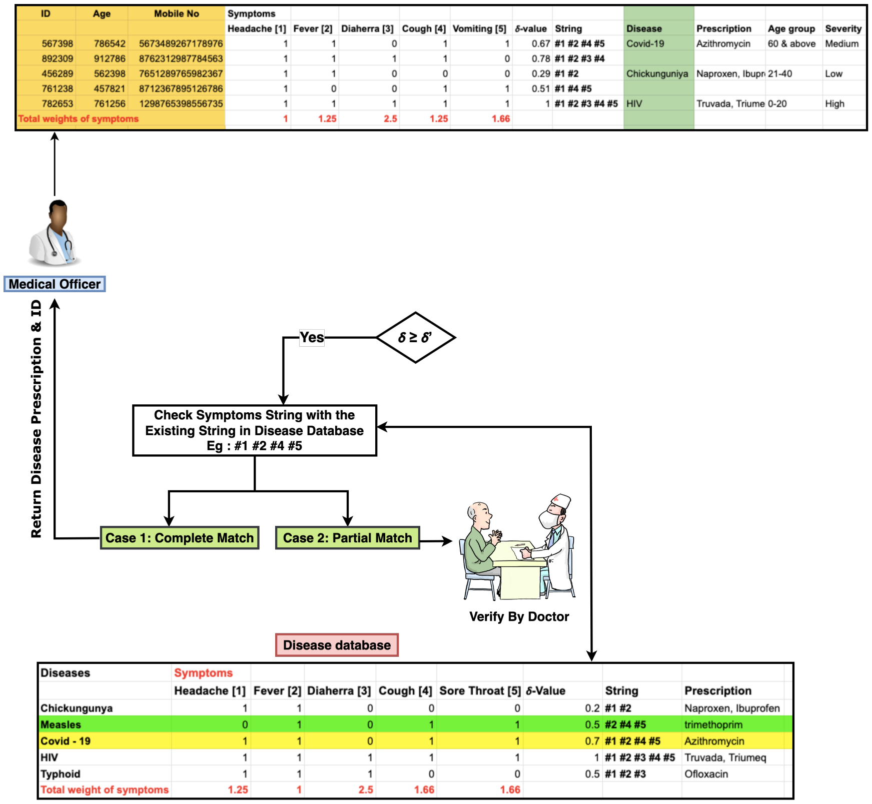


Figure 4.8: Sharing of data from medical officer to the CSP

Step 4: Medical officer sends δ-value, symptoms string and patient’s ID to the CSP. Based on the δ-value, the CSP will detect the disease as shown in Fig. 4.8.

Step 5: If $\delta \geq \delta'$, it will fall under the category ’yes’. The CSP will check the symptoms string with the string stored in the disease database. Based on the checking, there will be two scenarios.

Case 1: Fig. 4.9 shows a complete match of symptoms string in the diseases database. In this case, we will return the disease prescription to the medical officer with the patient’s ID.



ID	Age	Mobile No	Symptoms										
			Headache [1]	Fever [2]	Diaherra [3]	Cough [4]	Vomiting [5]	δ-value	String	Disease	Prescription	Age group	Severity
567398	786542	5673489267178976	1	1	0	1	1	0.67	#1 #2 #4 #5	Covid-19	Azithromycin	60 & above	Medium
892309	912786	8762312987784563	1	1	1	1	0	0.78	#1 #2 #3 #4				
456289	562398	7651289765982367	1	1	0	0	0	0.29	#1 #2	Chickunguniya	Naproxen, Ibupr	21-40	Low
761238	457821	8712367895126786	1	0	0	1	1	0.51	#1 #4 #5				
782653	761256	1298765398556735	1	1	1	1	1	1	#1 #2 #3 #4 #5	HIV	Truvada, Triume	0-20	High
Total weights of symptoms			1	1.25	2.5	1.25	1.66						

Medical Officer
Return Disease Prescription & ID
Yes
δ ≥ δ'
Check Symptoms String with the Existing String in Disease Database Eg : #1 #2 #4 #5
Case 1: Complete Match
Case 2: Partial Match
Verify By Doctor
Disease database

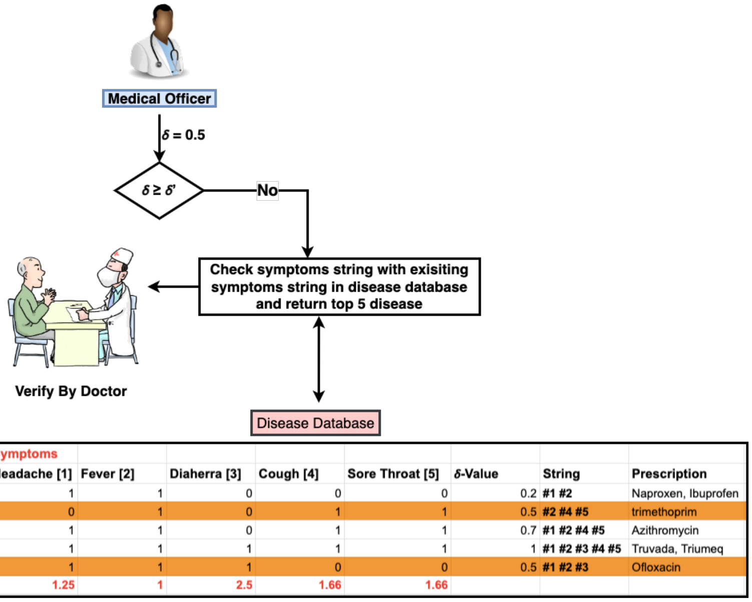
Diseases	Symptoms							
	Headache [1]	Fever [2]	Diaherra [3]	Cough [4]	Sore Throat [5]	δ-Value	String	Prescription
Chickungunya	1	1	0	0	0	0.2	#1 #2	Naproxen, Ibuprofen
Measles	0	1	0	1	1	0.5	#2 #4 #5	trimethoprim
Covid - 19	1	1	0	1	1	0.7	#1 #2 #4 #5	Azithromycin
HIV	1	1	1	1	1	1	#1 #2 #3 #4 #5	Truvada, Triumeq
Typhoid	1	1	1	0	0	0.5	#1 #2 #3	Ofloxacin
Total weight of symptoms	1.25	1	2.5	1.66	1.66			

Figure 4.9: Checking of complete/partial match in disease database

For instance, suppose that the medical officer sent symptoms string #1 #2 #4 #5 to the CSP. When the CSP checks this symptoms string with the string stored in the disease database, it finds a complete match with disease name covid-19 (yellow color row as shown in Fig. 4.9). The CSP will return the disease prescription and ID to the medical officer. The medical officer will update that disease prescription in the database. Based on the severity of the disease and age groups medical officer will send the prioritized patients to the medical staff. Later, the medical staff will decrypt the id and mobile number to send the relevant details to the patient.

Case 2: Fig. 4.9 shows a partial match of symptoms string in the disease database. In this case, we will choose the best string and send it to the doctor for further verification. For instance, suppose that the medical officer passes the symptoms string, i.e., #1 #4 #5, to the CSP when the CSP checks that symptoms string with the string stored in the disease database, instead of #1 #4 #5, it finds out #2 #4 #5. It will be a partial match (green row in the disease database in Fig. 4.9).

Step 6: However, if the $\delta \leq \delta'$ it will fall in a category 'no'. Then the CSP will check a symptom string with an existing symptoms string in the disease database. Based


Medical Officer
$\delta = 0.5$
$\delta \geq \delta'$
No
Check symptoms string with exisiting symptoms string in disease database and return top 5 disease
Verify By Doctor
Disease Database

Diseases	Symptoms							
	Headache [1]	Fever [2]	Diaherra [3]	Cough [4]	Sore Throat [5]	δ-Value	String	Prescription
Chickungunya	1	1	0	0	0	0.2	#1 #2	Naproxen, Ibuprofen
Measles	0	1	0	1	1	0.5	#2 #4 #5	trimethoprim
Covid - 19	1	1	0	1	1	0.7	#1 #2 #4 #5	Azithromycin
HIV	1	1	1	1	1	1	#1 #2 #3 #4 #5	Truvada, Triumeq
Typhoid	1	1	1	0	0	0.5	#1 #2 #3	Ofloxacin
Total weight of symptoms	1.25	1	2.5	1.66	1.66			

Figure 4.10: Selecting top 5 disease in decreasing order of symptoms weight

on decreasing the weight of symptoms, we will send the first top 5 highest weight diseases to the doctor for verification. For instance, in below Fig. 4.10, medical officer pass $\delta = 0.5$ that is less than δ'. In this case, the CSP will calculate the weight of the top 5 diseases with the same δ-value passed by the medical officer. Here, the weight of the disease is calculated as follows:

$$\begin{aligned} weight\ of\ measles\ disease =& \frac{wt(Fever) + wt(Cough) + wt(Sore\ Throat)}{Total\ weight\ of\ symptoms} \\ =& 4.32/8.07 \\ =& 0.5 \end{aligned} \tag{4.8}$$

Doctor:

Step 7: The doctor will diagnose the patients based on the detected symptoms and prioritize the patients based on their severity. For instance, as shown in Fig. 4.9 the severity of the patient's id no. 5 is high than that of id no. 3. In this case, the doctor will first examine id no. 5.

CHAPTER 5

RESULTS AND DISCUSSION

This chapter presents the results of the experiments conducted to validate the proposed methods.

5.1 Results of S-DIHE framework

We have used a dataset of 30 images as shown in Fig.5.1 where the images are in jpg format. Some manipulations such as colour constrat, copy move, removing some content from image are done in these images to create the near duplicate images. Some of these images from this dataset are shown in Fig.6, where the left is the original image and it's modified version is shown in the right. For instance, the computation of δ-value for one such image pair shown in Fig.7 is as follows:
Assuming image height = 300, image width = 300, block height = 10, and block width = 10

$$Total\ blocks = \frac{Image\ height \times Image\ width}{Block\ height \times Block\ width} \tag{5.1}$$

$$= \frac{300 \times 300}{10 \times 10} = 900 \tag{5.2}$$

$$\delta = \frac{No.\ of\ blocks\ with\ the\ same\ hash\ values}{Total\ no\ of\ blocks} \tag{5.3}$$

$$= \frac{785}{900} = 0.87 \tag{5.4}$$

The δ-values of the images can range from 0 to 1, with low to the high similarity between image pairs. The δ-value range from 0 to 0.30 can be considered as low similarity between image pairs, the range from 0.30 to 0.70 as medium similarity, and the range from 0.70 to 1 as high similarity.

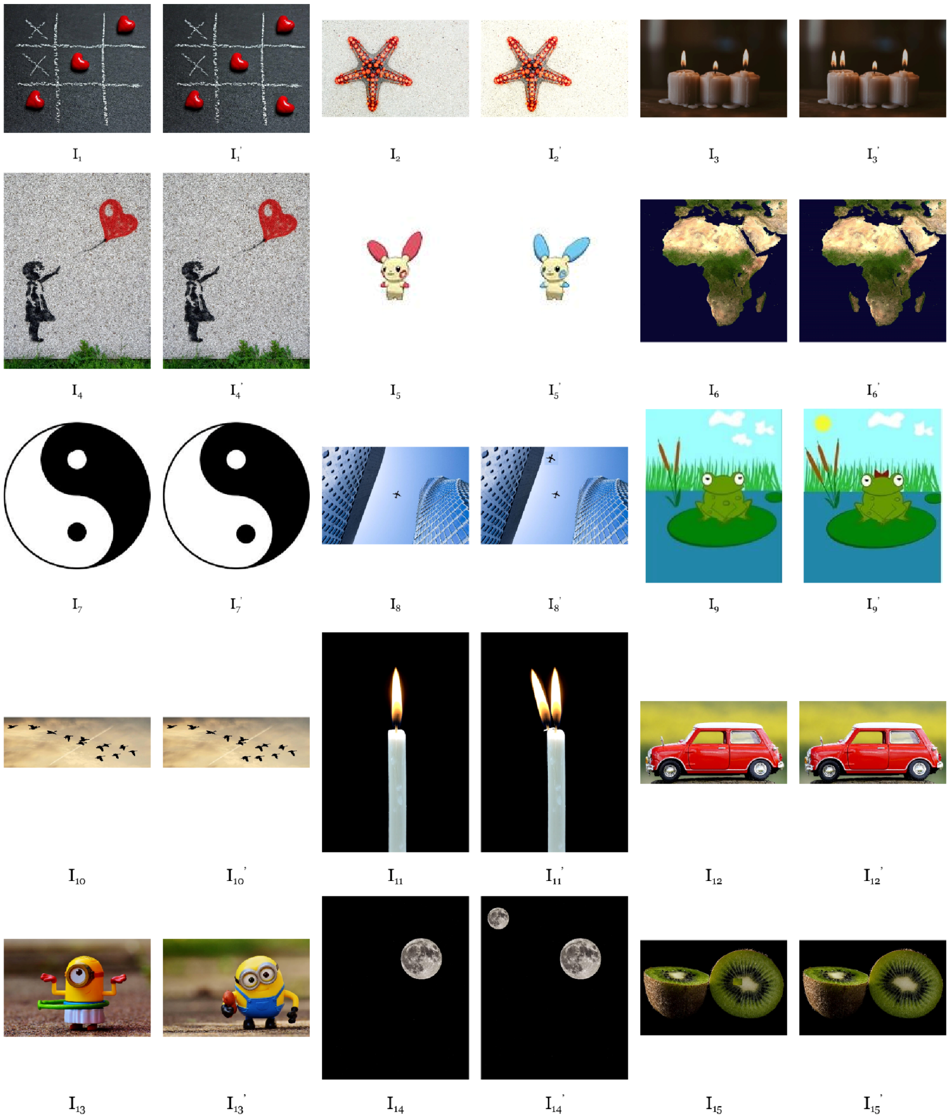


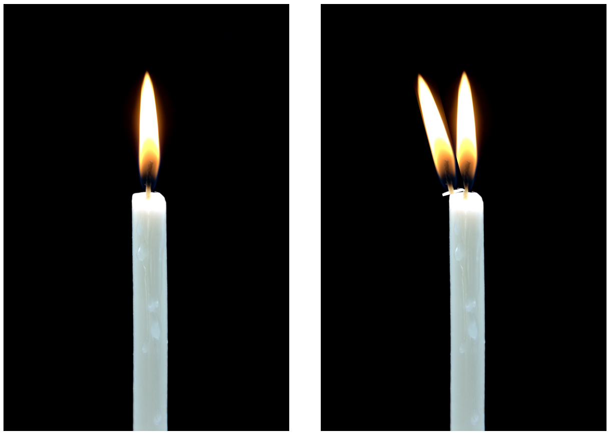
Figure 5.1: Near duplicate image pairs (I, I') with varying delta (δ)-value

For image pair in Fig.5.2, the δ-value = 0.87 implies highly similar images. This implies more scope for data deduplication. Here, the total size of the image pair is 372 KB which reduces to 134 KB storage space after deduplication. This is done for all image pairs, tabulated in Table 5.1 and depicted in Fig.5.3 where the storage (in KB) verses δ-values is plotted. In total, the space requirement for 30 images reduces from 5462 KB to 598 KB, saving 1864 KB.

In the second experiment, the total time spent for executing the protocol with various block sizes: 10x10, 20x20, and 30x30 was performed and shown in Fig.5.4

Table 5.1: Image pairs with delta (δ)-values.

Image Pair	Count of blocks with the same hash values	Total no. of blocks	δ-value (similarity)
$\delta(I_1, I_1')$	123	900	0.13
$\delta(I_2, I_2')$	33	900	0.03
$\delta(I_3, I_3')$	280	900	0.31
$\delta(I_4, I_4')$	471	900	0.52
$\delta(I_5, I_5')$	720	900	0.80
$\delta(I_6, I_6')$	600	900	0.66
$\delta(I_7, I_7')$	847	900	0.94
$\delta(I_8, I_8')$	240	900	0.26
$\delta(I_9, I_9')$	319	900	0.35
$\delta(I_{10}, I_{10}')$	101	900	0.11
$\delta(I_{11}, I_{11}')$	900	900	1
$\delta(I_{12}, I_{12}')$	785	900	0.87
$\delta(I_{13}, I_{13}')$	0	900	0
$\delta(I_{14}, I_{14}')$	253	900	0.28
$\delta(I_{15}, I_{15}')$	114	900	0.12



(a) (b)

Figure 5.2: (a) original image and (b) modified image

Based on the experiment, it is observed that execution time for smaller blocks is longer than for larger blocks but better deduplication can be achieved with smaller block size.

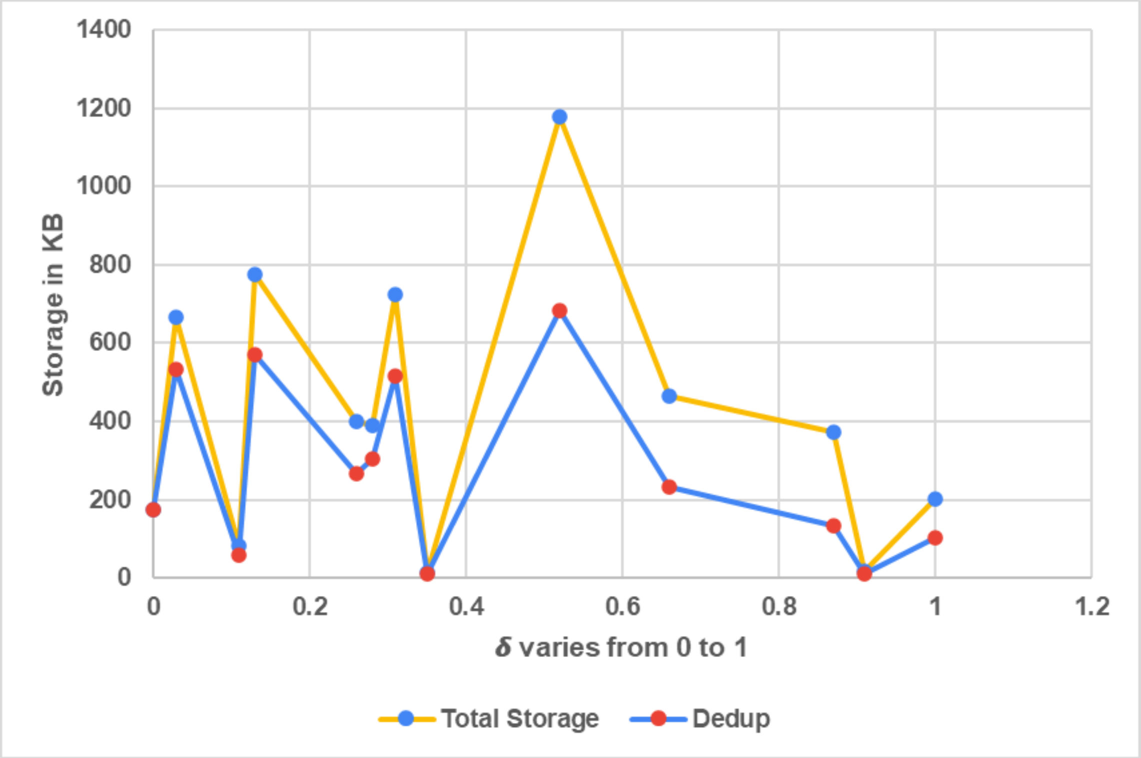


Figure 5.3: Variation of delta (δ) from 0 to 1

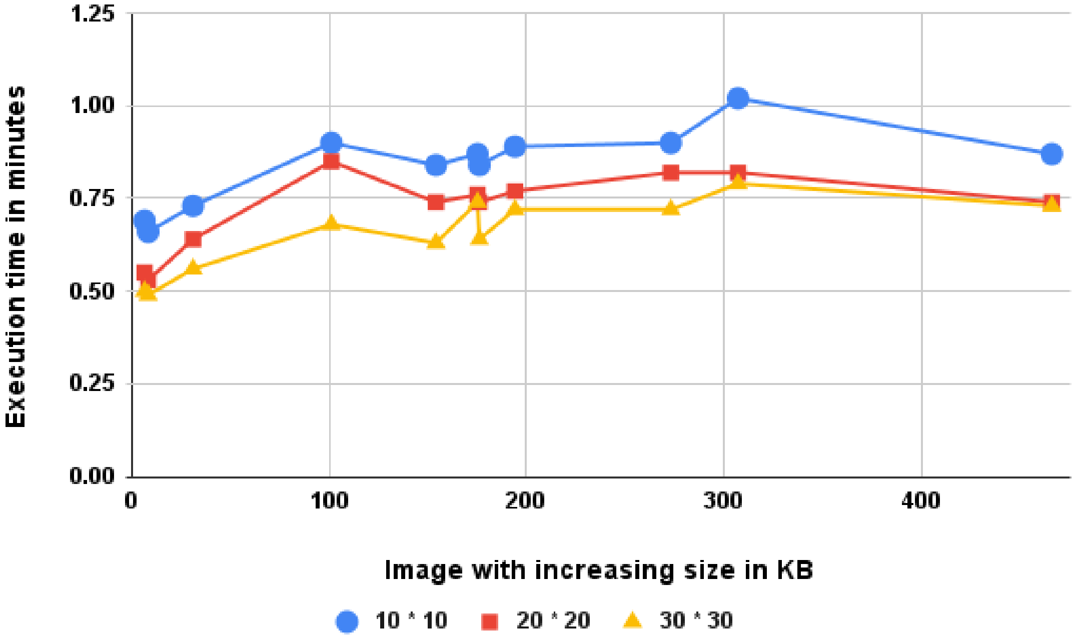


Figure 5.4: Comparison of execution times for various block sizes

5.2 Results of Privacy-Preserving Disease Prediction and Secure Data Deduplication of Health Data

This section describes the results of our experiment to validate the proposed method.

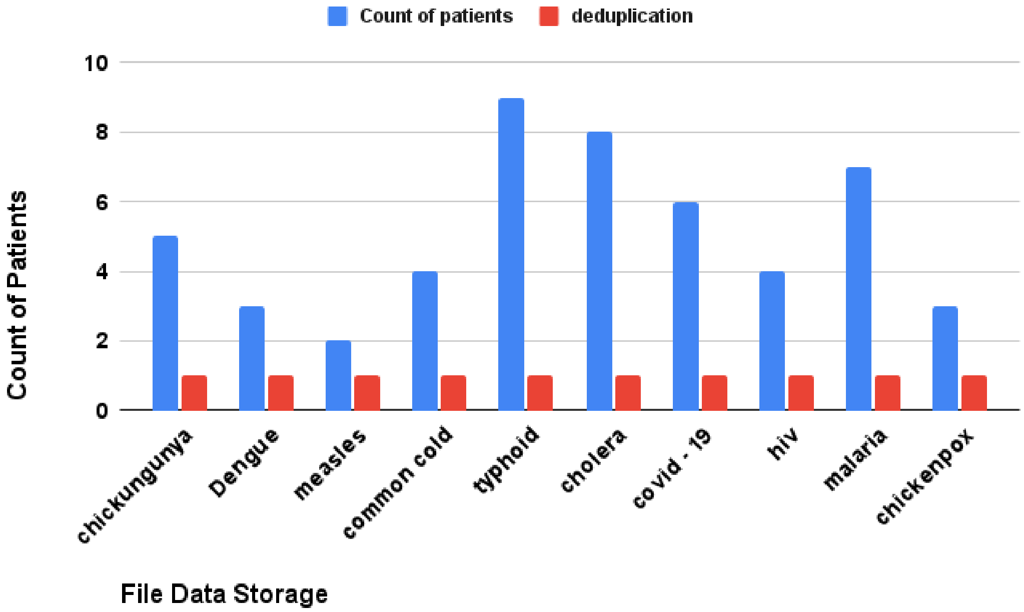


Figure 5.5: Data storage of the file containing the disease and its corresponding prescription

In our result, the file contains the disease and its corresponding prescription. The size of each file is 10 kb. Before deduplication, the storage space required for uploading 10 files will be 510 kb, which might also consist of duplicate files. We save 410 kb of storage space after performing deduplication. It removes duplicate

CHAPTER 6

Security Analysis

In this chapter, we have analyzed the security of the proposed S-DIHE and Privacy-Preserving Disease Prediction and Secure Data Deduplication of Health Data.

6.1 Analysis of S-DIHE

Lemma 1: *Block-level deduplication is secured against poison attacks.*
Proof: The attacker's aim is to access the user's file. It first computes the duplicate ciphertext (C) and tag (T) as shown below:

$$C^* \leftarrow E(K^*, I^*)$$
$$T^* \leftarrow H(C^*)$$

The user computes $C \leftarrow E(K, I)$, $T \leftarrow H(C)$. It sends T to the CSP. If $T = T^*$ then the user will end up downloading fake C^* instead of C. The user receives C^* and recomputes $T^{''}$ as $T^{''} \leftarrow H(C^*)$. However, if $T^{''} \neq T$, then integrity check on the user's end fails. Therefore the user considers the file has been changed and updates the CSP. Hence, S-DIHE is secure against poison attack.

Lemma 2: *Block-level deduplication is secured against frequency analysis attacks.*
Proof: In the frequency analysis attack, an attacker can map the ciphertext to the original message. We have used paillier encryption algorithm. It is a probabilistic encryption scheme that breaks the mapping between original data and its corresponding ciphertext. Hence, S-DIHE is secure against frequency analysis attacks.

Lemma 3: *Block-level deduplication is secure against side-channel attacks.*
Proof: In the case of a side-channel attack, an attacker wants to know the presence of the data and its content by guessing tag values stored at the CSP. For instance, an attacker's first attempt at guessing the stored tag value is T^*, where $T \neq T^*$.

Here T is the tag value stored at the CSP. From the above scenario, there will be two cases, either the data is not present on the CSP with tag T^* or the guessed value of the tag is wrong. The attack is restricted by the rate of the limiting factor k. When an attacker makes repeated attacks it is unable to known the content and the presence of the data which secures the system's privacy. Hence, S-DIHE is secure against side-channel attack.

Lemma 4: *Block-level deduplication is secure against dictionary attacks.*
Proof: An attacker wants to gain ownership of data, so, it computes a duplicate tag value T^*. On receiving the T^* from an attacker, the CSP perform the search operation in its tag table. If $T == T^*$, then the CSP will ask the user to prove its authenticity by answering the challenge before providing the data. The CSP generates a challenge $\pi c'$ (last block of cipher image C) and ask the user to send πc (last block of cipher image C). If these two blocks, πc and $\pi c'$, gets match, then the CSP responds back to the user and provide data pointer. Hence, S-DIHE is secure against dictionary attack.

6.2 Analysis of Privacy-Preserving Disease Prediction and Secure Data Deduplication of Health Data

Lemma 5: The privacy of the patient is preserved in the proposed framework.
Proof: In the proposed framework, the patient provides the necessary clinical symptoms and personal information to the medical staff who is assumed to be an honest entity. Prior to sending the patient's information further to the medical officer and CSP, personal data of the patient is encrypted using the Paillier homomorphic encryption. Further processing by the medical officer is done based on the symptoms string to obtain the δ value that is sent to the CSP for disease prediction. Once the disease is predicted, only then the entire data is sent back to the medical staff that decrypts the patient's ID and communicates the diagnoses result to the patient. At no point in the entire process, any unauthorized user can access the personal information of the patient. Hence, preserving the patient's privacy in the proposed framework.

CHAPTER 7
Conclusion

This thesis proposed a secure block-level data deduplication scheme exploiting the robust homomorphic image hashing. The scheme assured dual side data integrity check and proved secure against specific attack scenarios i.e. poison attacks, dictionary attacks, side-channel attacks, and frequency analysis attacks. In the future, study can be conducted to theoretically analyse the threshold values for different kinds of images to declare them as similar or not based on their content.

We also proposed a privacy-preserving framework to perform disease prediction. The patient's personal details are encrypted using a paillier homomorphic encryption scheme, and no data is leaked at any point in the entire process. Also, to optimize the space requirements, secure data deduplication is done to prescribe the disease. The proposed framework is focused on supporting the initial diagnosis of the diseases based on the patient's clinical symptoms and prioritizing the appointment queue of the doctor based on the severity of the disease and the age groups the patient falls into. In future, we wish to extend this work to more severe diseases that have supporting data, such as CT scan images or Magnetic Resonance Imaging (MRI) scans that have more scope for data deduplication.

References

[1] Overland Storage, NAS Appliances, Blades Storage, Internal DAS Storage, Internal Shared Storage, Direct Connect SAS, Tape Blades, Fibre Channel Storage, and Solutions Storage. Data deduplication, 2008.

[2] Wen Xia, Hong Jiang, Dan Feng, Fred Douglis, Philip Shilane, Yu Hua, Min Fu, Yucheng Zhang, and Yukun Zhou. A comprehensive study of the past, present, and future of data deduplication. *Proceedings of the IEEE*, 104(9):1681–1710, 2016.

[3] Huaglory Tianfield. Security issues in cloud computing. In *2012 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1082–1089. IEEE, 2012.

[4] Zheng Yan, Mingjun Wang, Yuxiang Li, and Athanasios V Vasilakos. Encrypted data management with deduplication in cloud computing. *IEEE Cloud Computing*, 3(2):28–35, 2016.

[5] Chun-I Fan, Shi-Yuan Huang, and Wen-Che Hsu. Encrypted data deduplication in cloud storage. In *2015 10th Asia Joint Conference on Information Security*, pages 18–25. IEEE, 2015.

[6] Zheng Yan, Wenxiu Ding, and Haiqi Zhu. A scheme to manage encrypted data storage with deduplication in cloud. In *International Conference on Algorithms and Architectures for Parallel Processing*, pages 547–561. Springer, 2015.

[7] N Jayapandian and AMJ Md Zubair Rahman. Secure deduplication for cloud storage using interactive message-locked encryption with convergent encryption, to reduce storage space. *Brazilian Archives of Biology and Technology*, 61, 2018.

[8] Ashish Agarwala, Priyanka Singh, and Pradeep K Atrey. Client side secure image deduplication using dice protocol. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 412–417. IEEE, 2018.

[9] Amit Pandey and Gyan Prakash. Deduplication with attribute based encryption in e-health care systems. *International Journal of MC Square Scientific Research*, 11(4):16–24, 2019.

[10] R Shiny Sharon and R Joseph Manoj. E-health care data sharing into the cloud based on deduplication and file hierarchical encryption. In *2017 International Conference on Information Communication and Embedded Systems (ICICES)*, pages 1–6. IEEE, 2017.

[11] Priyanka Singh and Hany Farid. Robust homomorphic image hashing. In *CVPR Workshops*, pages 11–18, 2019.

[12] B Rasina Begum and P Chitra. Seeddup: a three-tier secure data deduplication architecture-based storage and retrieval for cross-domains over cloud. *IETE Journal of Research*, pages 1–18, 2021.

[13] Basappa B Kodada and Demian Antony D'Mello. Secure data deduplication (sdˆ 2edup) in cloud computing: Threats, techniques and challenges. In *Advances in Communication and Computational Technology*, pages 1239–1251. Springer, 2021.

[14] B Tirapathi Reddy, P Sai Kiran, T Priyanandan, CH Vikas Chowdary, and B Jaya Aditya. Block level data-deduplication and security using convergent encryption to offer proof of verification. In *2020 4th International Conference on Trends in Electronics and Informatics (ICOEI)(48184)*, pages 428–434. IEEE, 2020.

[15] Liang Wang, Baocang Wang, Wei Song, and Zhili Zhang. A key-sharing based secure deduplication scheme in cloud storage. *Information Sciences*, 504:48–60, 2019.

[16] Mihir Bellare and Sriram Keelveedhi. Interactive message-locked encryption and secure deduplication. In *IACR international workshop on public key cryptography*, pages 516–538. Springer, 2015.

[17] Jin Li, Xiaofeng Chen, Mingqiang Li, Jingwei Li, Patrick PC Lee, and Wenjing Lou. Secure deduplication with efficient and reliable convergent key management. *IEEE transactions on parallel and distributed systems*, 25(6):1615–1625, 2013.

[18] Mihir Bellare, Sriram Keelveedhi, and Thomas Ristenpart. Message-locked encryption and secure deduplication. In *Annual international conference on*

the theory and applications of cryptographic techniques, pages 296–312. Springer, 2013.

[19] Phaneendra Kanakamedala, Abburi Akhil, Triveni Buska, Kandimalla Koteswara Rao, and Mallela Sireesha. Attribute-based storage supporting secure deduplication of encrypted data in cloud. In *Proceedings of Third International Conference on Intelligent Computing, Information and Control Systems*, pages 81–89. Springer, 2022.

[20] Ashish Agarwala, Priyanka Singh, and Pradeep K Atrey. Dice: A dual integrity convergent encryption protocol for client side secure data deduplication. In *2017 IEEE international conference on systems, man, and cybernetics (SMC)*, pages 2176–2181. IEEE, 2017.

[21] Han Gang, Hongyang Yan, and Lingling Xu. Secure image deduplication in cloud storage. In *Information and Communication Technology-EurAsia Conference*, pages 243–251. Springer, 2015.

[22] Lu Chen, Feng Xiang, and Zhixin Sun. Image deduplication based on hashing and clustering in cloud storage. *KSII Transactions on Internet and Information Systems (TIIS)*, 15(4):1448–1463, 2021.

[23] Naga Raju Hari Manikyam and Munisamy Shyamala Devi. An image decompression model with reversible pixel interchange decryption model using data deduplication. *Traitement du Signal*, 39(1):195–203, 2022.

[24] Ata Ullah, Khubab Hamza, Muhammad Azeem, and Fadi Farha. Secure healthcare data aggregation and deduplication scheme for fog-orineted iot. In *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*, pages 314–319. IEEE, 2019.

[25] AV Usharani and Girija Attigeri. Secure emr classification and deduplication using mapreduce. *IEEE Access*, 10:34404–34414, 2022.

[26] Jinsu Kim, Sungwook Ryu, and Namje Park. Privacy-enhanced data deduplication computational intelligence technique for secure healthcare applications. *CMC-COMPUTERS MATERIALS CONTINUA*, 70(2):4169–4184, 2022.

[27] Marci Meingast, Tanya Roosta, and Shankar Sastry. Security and privacy issues with health care information technology. In *2006 International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 5453–5458. IEEE, 2006.

[28] Jie Xu, Kaiping Xue, Shaohua Li, Hangyu Tian, Jianan Hong, Peilin Hong, and Nenghai Yu. Healthchain: A blockchain-based privacy preserving scheme for large-scale health data. *IEEE Internet of Things Journal*, 6(5):8770–8781, 2019.

[29] Sebastian Haas, Sven Wohlgemuth, Isao Echizen, Noboru Sonehara, and Günter Müller. Aspects of privacy for electronic health records. *International journal of medical informatics*, 80(2):e26–e31, 2011.