Homomorphic-based Searching Scheme for Cloud Connected Devices

MCS

By

Tayyaba Anwer

A thesis submitted to the faculty of Information Security Department, Military College of Signals, National University of Sciences and Technology, Rawalpindi in partial fulfilment of the requirements for the degree of MS in Information Security

April 2023

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Tayyaba Anwer

April 2023

Dedication

This thesis is dedicated to my family, teachers, and friends

for their unconditional love, endless support, and continuous encouragement.

Acknowledgement

All worship and glory be to the All Magnificent and All-Merciful Allah Almighty. I am thankful to all my family and particularly my parents for they have always been my pillar of utmost strength and assistance.

I pay humble gratitude to my Project Supervisor Dr. Sahazaib Tahir who not only supervised my research but for mentoring me in a very polite yet considerate and helpful manner. As a supervisor, his facilitation and counseling have always been an irreplaceable resource of guidance for me and will continue to be so in the coming years of my life. I am also grateful to my worthy co-supervisor Dr. Fawad khan for his highly helpful comments which helped me refurbish my skills and bring a refined edge to this research as well as to my esteemed committee members for their guidance, support, and suggestions.

I am deeply grateful to my colleague PhD scholar Aiman Sultan who has guided me wholeheartedly, helped me overcome my limitations and shortcomings, and motivated me whenever this objective seemed not achievable. Lastly, I am grateful to the National University of Sciences and Technology (NUST) and my batch mates for giving me healthy competition and countless memories to cherish forever.

Abstract

Surveillance data images of crowded places have become increasingly important in today's world for a variety of reasons. Cloud-based image searching schemes leverage distributed computing power to search large amounts of data quickly and accurately, allowing users to easily find relevant images based on their search queries. Multiple schemes have been proposed but there remain issues related to the security and privacy of data owners as well as the revelation of search and access patterns. Moreover, existing homomorphic techniques for encrypted data are inefficient enough to be deployed in a real-world environment. A novel scheme for similar image searching based on homomorphic encryption is proposed for cloud-connected devices. The scheme is based on probabilistic trapdoors and ensures the privacy preservation of image data. The scheme also provides search pattern security and is CPA-secure in an adversarial model. The scheme also fulfills the property of Query-Trapdoor and Trapdoor-Image Indistinguishability. Implementation and testing of the proposed scheme are carried out over a real-world data set in order to assess its security and performance in terms of complexity, computation, and storage overheads.

Contents

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

AES	Advanced Encryption Standard
HE	Homomorphic Encryption
RSA	Rivest–Shamir–Adleman
SE	Searchable Encryption
LBES	Lattice-based Encryption Schemes
ZKP	Zero Knowledge Proof
CS	Cloud Storage / Server
CSP	Cloud Service Provider
SSE	Symmetric Searchable Encryption
PEKS	Public Key Encryption with Keyword Search

PE	Predicate encryption
PIE	Inner Product Encryption
IBE	Identity-Based Encryption
MRSE	Multi-keyword Ranked Search Encryption
PIR	Private Information Retrieval
PHE	Partial Homomorphic Encryption
SHE	Somewhat Homomorphic Encryption
FHE	Fully Homomorphic Encryption
SIFT	Scale-Invariant Feature Transform
CSD	Colour Structure Descriptor
CLD	Colour Layout Descriptor
HOG	Histogram of Oriented Gradients
HM	Hahn Moment
PPHE	Privacy-Preserving Homomorphic Encryption
DCT	Discrete Colour Transform
DWT	Discrete Wavelength Transform
CBIR	Content-based Image Retrieval
EBCBIR	Efficient and Privacy-Preserving Content-based Image Retrieval

LBP	Linear Binary Pattern
MSB	Most Significant Bit
XOR	Exclusive-OR
EHD	Edge Histogram Descriptor
LSD	Locality Sensitive Hashing
kNN	k-Nearest Neighbor
SMH	Secure Modular Hashing
BoW	Bag of Words
R-CNN	Region-based Convolutional Neural Networks
MTCNN	Multi-Task Cascaded Convolutional Neural Networks
YOLO	You Only Look Twice
FSAF	Feature Selective Anchor Free
DETR	Detection Transformer
mAP	mean average precision
FPS	frames per second
PS	proposed scheme

Symbols

CS	cloud server
p,q	prime numbers
pb	prime bits
μ	modular multiplicative inverse
λ	LCM *(p-1,q-1)*
gp	*getprime* function
RN()	function returning random number
s_k	private (secret) key
p_k	public key
I_i	Image
E_{I_i}	Encrypted Image
p()	*power* function
F_i	facial image
H_{F_i}	facial image hash
$E_{H_{F_i}}$	encrypted facial hash
Q_I	query image
H_{Q_I}	query image hash

T_Q	trapdoor
T'_Q	additive inverse of trapdoor
V_R	resultant vector with searching results
V_{th}	threshold value for similarity
A_{DD}	addition function
E_s	AES encryption
D_s	AES decryption
$h()$	hash operation

Chapter 1

Introduction

1.1 Overview

Surveillance data images of crowded places have become increasingly important in today's world for a variety of reasons. In the first place, they are very useful in preventing and investigating criminal activity, such as theft, vandalism, and violence. In crowded places, cameras and surveillance data can serve as a deterrent to potential criminals, thus reducing crime rates [1]. In addition to their use in criminal investigations, surveillance data images can also provide valuable insights into crowd behavior, enabling authorities to detect potential safety hazards or emergency situations before they escalate. This is particularly important in crowded places such as train stations, airports, and sports arenas, where large numbers of people gather in confined spaces [2]. Facial recognition in such surveillance systems plays a vital role in carrying out security checks against suspicious personnel. This is possible nowadays due to installments of intelligent cameras

and smart surveillance sensors [3].

Surveillance data images can also play a critical role in post-incident investigations. In the event of a security breach, accident, or emergency, all the data that has been gathered by these surveillance / smart cameras can be used to reconstruct events, identify potential suspects or witnesses, and help authorities to develop strategies for preventing similar incidents from occurring in the future [4]. Moreover, the information derived from surveillance data images can be used for multiple applications; for instance, traffic management, crowd control, and urban planning. By analyzing the patterns of crowd behavior, traffic flow, and other factors, authorities can make informed decisions about the design and management of public spaces, enhancing the safety and well-being of citizens [5].

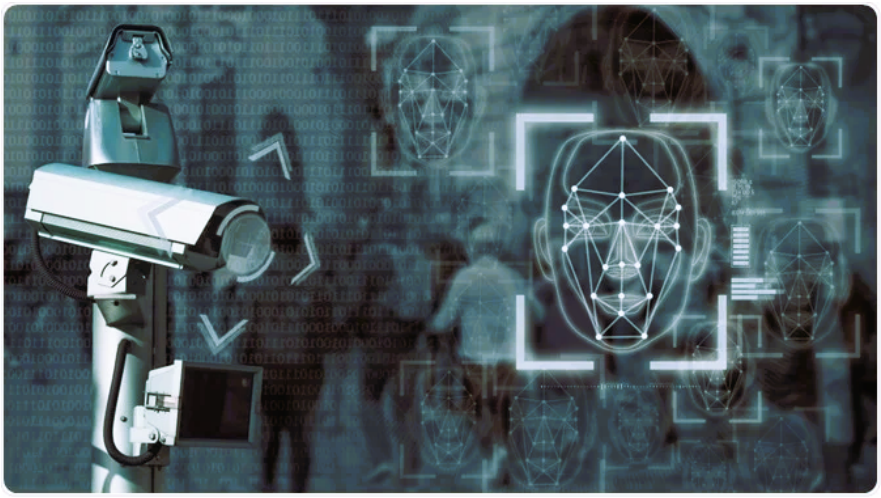


Figure 1.1: Facial Recognition in Intelligent Surveillance Cameras

The data being used for crowd analysis includes facial detection and then attribute extraction from those facial images as depicted by figures 6.1 (a) and (b) respectively. The data generated by these devices is so enormously growing each day that its local storage, management, and security is an issue in itself. With the advance-

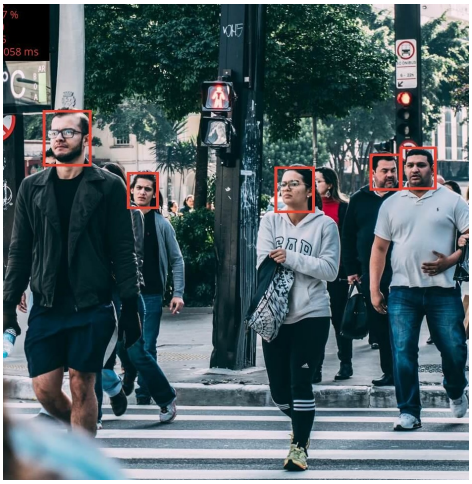
(a)  (b) 

Figure 1.2: (a) Face Detection (b) Facial Attributes Markings

ments in technology as well as digitization of data, an increase in the necessity for extra storage capacity has led to the development of multiple third-party storage servers *i.e.* cloud services, etc. Cloud Server Providers (CSP) provide storage as well as different processing facilities to individual users as well as enterprises where it can be accessed worldwide over internet connectivity [6].

Cloud storage and searching schemes are crucial for the efficient and scalable management of images stored over the cloud. Cloud storage presents users with the benefits of on-demand scalable resources and accessibility on a pay-per-use model. Cloud storage enables users to store large amounts of data without worrying about storage limitations and can also provide automated backups, ensuring data security [7]. In addition, cloud searching schemes make it easier to search and retrieve images stored in the cloud. Cloud-based image searching schemes leverage distributed computing power to search large amounts of data quickly and accurately, allowing users to easily find relevant images based on their search queries [8]. Moreover, cloud searching schemes can also perform image analysis tasks, such

as object detection/recognition, and content-based image retrieval/classification, which are essential for various applications, including e-commerce, social media, and healthcare. Overall, cloud storage and searching schemes play a vital role in managing and retrieving images stored over the cloud, facilitating fast and efficient access to image data, and enabling innovative applications that rely on image analysis and search.

However, outsourcing private data poses a threat to confidentiality as the cloud is termed as *honest but curious* entity. Another main concern is that many cloud service providers exploit their customers' personal information for marketing and other business objectives. Therefore, to address such issues, the data owners encrypt their data before outsourcing it to Cloud [9]. Many such encryption techniques have a number of drawbacks [10] as well as require multiple searchable encryption schemes [11]. Therefore, it is impossible for a data owner to work on his cloud data without retrieving it, decrypting it at his end, processing it as needed, and then encrypting it again in order to return it to the cloud. In addition to that, additional processing is needed on both the user/owner's side, which results in resource usage and time consumption. A lot of searchable encryption techniques are in use allowing a user to search over encrypted data without the need to disclose any information about the underlying plaintexts [12].

A basic searchable encryption model is shown in figure 1.3. It contains 3 major entities *i.e.,* data owner, user(s), and a cloud server. The data owner encrypts the data and outsources it to the cloud server. Any authorized user generates a query requesting certain data from the cloud server. In response, the cloud server

retrieves the requested data and sends that to the user (also known as the client). The user, in this case, can be the data owner who has outsourced the data and it can also be some other legitimate user who can access the data at any time provided he has the required resources. The data can be a document or image or any other media file.

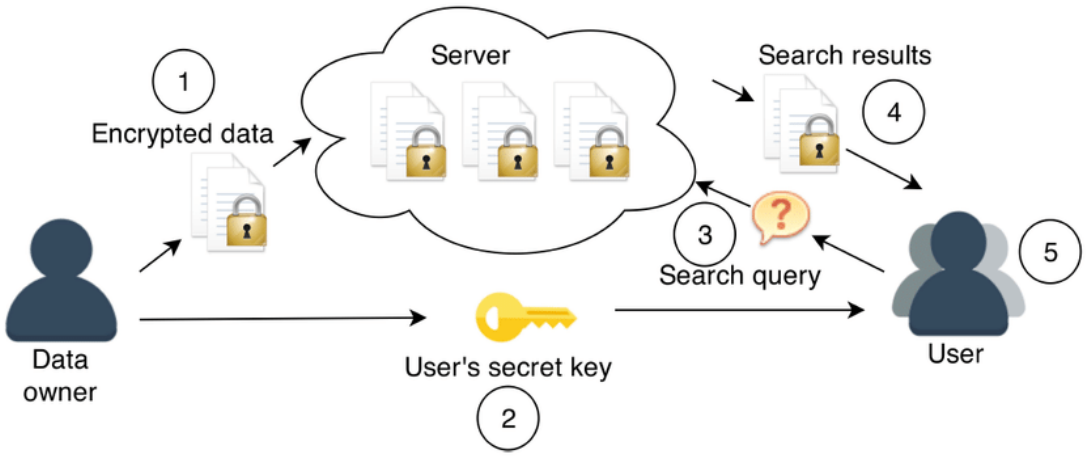


Figure 1.3: Searchable Encryption Model

As the use of cloud computing becomes more prevalent, the need for efficient and secure image retrieval methods over the cloud has become increasingly important. However, the transmission and storage of sensitive visual information, such as personal images, raises significant privacy concerns. In this research, we propose a novel privacy-preserving image-based homomorphic approach for similar image retrieval over the cloud. Our approach utilizes homomorphic encryption techniques to encrypt the visual data, allowing for secure transmission and storage while preserving the privacy of the users. Additionally, our approach employs an efficient image-matching algorithm, enabling accurate and fast retrieval of similar images. Through extensive experiments and comparisons with existing methods,

we demonstrate the efficiency and security of our proposed scheme for privacy-preserving similar image retrieval over the cloud.

1.2 Motivation

Cloud storage of surveillance data is a great way to ensure the security and accuracy of important records. By storing this information in the cloud, it can be accessed quickly by authorized personnel from any location with an internet connection. This makes it easier for authorized users to access critical evidence when needed, while also providing additional layers of security that are not available with traditional methods such as paper filing systems or on-site servers. Additionally, cloud storage allows for automated backups which ensure that all data remains safe even if there is a power outage or other emergency situation. Finally, using cloud storage helps reduce costs associated with maintaining physical infrastructure and provides greater scalability so organizations can easily expand their capacity as needed without having to invest in new hardware or software solutions.

This topic is contemporary in its field with the number of users of cloud-connected devices on an explosive rise day by day. An enormous amount of data is not only generated but also linked up to many cloud storage(s) where the issue of its privacy and security still poses a threat. The data from these devices need to be outsourced in encrypted form so that the data is not accessible to those with no authority over it. Along with it, the capability to conduct a search over it should also be

enabled so that the data is accessible easily without the need to download and decrypt all the data locally. Hence, all searching should be conducted over data that has been encrypted prior to outsourcing.

Multiple schemes have been proposed but there remain issues related to the security and privacy of data owners as well as the revelation of search and access patterns. Moreover, existing homomorphic techniques for encrypted data are inefficient enough to be deployed in a real-world environment. As a result, the ability to efficiently search over encrypted data using lightweight homomorphic encryption techniques, together with the necessity for huge data storage in the cloud, serves as the motivation for this research.

1.3 Advantages and Applications

As the explosive rise is being observed every day in the field of cloud computing, artificial intelligence, and generation of big data, the data owners are now shifting towards cloud services which reduces the storage and computational overhead for resource-constraint devices at the user end. These services are still not getting attention due to security concerns. Users care about the privacy of their data being shared and stored outside their jurisdiction. Fortunately, data processing in the encrypted domain can overcome this issue and provides different privacy-preserving techniques. Data processing in encrypted domain techniques already proposed by researchers is not secure for users in terms of privacy-preserving as approximately all these techniques do the deterministic query searching for batch

search. In this thesis, we will propose a novel technique that will be capable of a probabilistic query searching using homomorphic encryption. This will enhance the privacy of users in terms of search pattern leakage attacks. It also gives users privileges to control access to their data.

The proposed research has multiple applications where we have big data to handle and confidentiality is a crucial parameter. These vast regions of applications include but are not limited to:

- The industry of vehicle automation is much broader than a commercial car setup. As a result of the variety and utility of applications, automated systems have become increasingly popular with a clear growth market for vehicles / self-driving vehicles.
- E-healthcare-based medical setups using cloud services for maintenance of their medical data records.
- IoT-based organizations using cloud services for outsourcing their data.
- Crowded places *e.g.,* airports, railway stations, parking lots, etc. where security cameras are deployed and connected to cloud services for constant monitoring and storage.
- Multiple government and private organizations use cloud services for their outsourced data.

The applications of homomorphic encryption lie in sectors of finance, healthcare, the military, and different governmental bodies. There are different stakeholders

in every industry as these deal with sensitive data on a daily basis that is to be kept confidential in storage as well as secured in its communication. For some applications in military intelligence, homomorphic encryption (HE) can assist to balance threats and usefulness in information and data exchange. In summary, many corporations' data privacy and utility concerns, particularly sensitive ones, now need unpalatable trade-offs, which can have terrible consequences for both corporations and their contributors. Homomorphic encryption schemes potentially offer a unique solution at a low cost relative to the potential consequences [13]. Many such organizations now outsource their data to cloud storage services where secrecy and privacy are major concerns.

1.4 Problem Statement

With the widespread use of smart devices and the growth of mobile applications, a significant amount of image data is being generated daily. However, due to the limited storage capacity of end-user devices, data needs to be stored externally, which is often provided by cloud service providers (CSPs). CSPs offer not only storage but also the capability of searching and processing data over the internet, without restrictions based on user geo-location or jurisdiction. However, outsourcing data to CSPs raises privacy concerns as most CSPs do not offer encryption as a service, and those that do, lack techniques for privacy-preserving image retrieval. Existing searching techniques are not secure enough to protect user privacy. Many such schemes are deterministic in nature and thus, prone to

search pattern attacks. Moreover, most image processing and retrieval schemes do not support content-based image retrieval. Therefore, there is a need for an image retrieval technique that can ensure user privacy for batch queries, provide probabilistic searching over the cloud, and support fuzzy searching as well as retrieval. This thesis aims to address these challenges by focusing on probabilistic searching and retrieval scheme for similar images stored over the cloud server.

1.5 Research Methodology

The research methodology for a homomorphic encryption-based searching scheme for cloud-connected images is divided into different stages. Firstly, a comprehensive literature review of existing homomorphic encryption-based image searching schemes will be conducted which will help to identify the limitations of existing schemes and identify opportunities for improvement. The system architecture will be designed, and the requirements for the proposed homomorphic encryption-based image-searching scheme will be identified. The system model will be designed to ensure that it meets the requirements for secure and efficient image searching over the cloud. After that, a homomorphic encryption-based image-searching scheme will be designed that satisfies the requirements identified in the previous stage. The scheme will use advanced cryptographic techniques to ensure that the user's privacy is protected while allowing efficient searching of encrypted image data over the cloud. Lastly, the proposed scheme will be implemented, and its performance will be evaluated through various metrics, including efficiency, ac-

curacy, and security. The results will be compared with existing image-searching schemes, and the advantages and limitations of the proposed scheme will be identified. The scheme will be tested on a cloud computing platform to ensure that it can handle large amounts of data and support concurrent user queries. Finally, the thesis will conclude with a discussion of the proposed scheme's limitations and some future directions.

1.6 Research Objectives

The main contributions of this study are:

- A comprehensive literature review and analysis of object detection algorithms and existing privacy-preserving image-based searchable encryption techniques for cloud-connected devices.
- A novel scheme for similar image searching based on homomorphic encryption is proposed for cloud-connected devices. The scheme is based on probabilistic trapdoors and ensures the privacy preservation of image data.
- Implementation and testing of the proposed scheme are carried out over a real-world data set in order to assess its security and performance in terms of complexity, computation, and storage overheads.

1.7 Thesis Organization

The thesis is organized into the following chapters as follows:

- **Chapter 1:** This chapter provides a brief overview of the subject, discusses the motivation for this research, discusses some applications, presents the problem statement, explains the research goals and approach, and finally summarizes the contributions.

- **Chapter 2:** The chapter provides an introduction to searchable encryption and its types. It also presents homomorphic encryption and discusses some preliminaries. A comprehensive discussion on object detection algorithms and some of the latest existing SE schemes are put forward and a comparative analysis is carried out of some latest state-of-the-art SE schemes over the cloud. Lastly, some challenges and design goals are presented for a searchable encryption scheme over cloud storage.

- **Chapter 3:** The chapter presents the system model with security assumptions, security goals, and threat model.

- **Chapter 4:** The chapter presents the proposed scheme with all its phases in detail.

- **Chapter 5:** The chapter revisits security definitions and presents the security analysis of the proposed scheme.

- **Chapter 6:** The chapter presents the performance analysis of the proposed scheme in terms of computational complexity and storage overhead.

- **Chapter 7:** This chapter concludes the research.

Chapter 2

Literature Review

2.1 Overview

The chapter presents an overview of searchable encryption schemes and their types. It also discusses homomorphic encryption and briefly explains the Paillier Homomophic cryptosystem. Some basic preliminaries are discussed. A comprehensive literature review is carried out for existing SE schemes as well as object detection algorithms. Lastly, a comparison is drawn among the latest image-based SE schemes for cloud-connected devices.

2.2 Searchable Encryption

Over the past few years, the exponential digitization of data and its processing have given birth to many issues with respect to data storage, maintenance, and availability of required resources to carry out these functionalities. Third-party

storage mediums provided a solution for the above-mentioned issues [14]. However, the cloud has been unanimously termed as *'honest but curious'*, and different security measures are required to ensure the privacy of data before outsourcing it to cloud storage (CS). In simple terms, a data owner wants to outsource data but not before encrypting or securing it from any threat from the cloud. Since the data is stored in encrypted form on the cloud, at the user's end, encrypted queries have to be generated and all the processing and searching has to be carried out over encrypted data [15]. These different searching techniques are termed Searchable Encryption (SE) [16]. Description, merits, and demerits of different types of searchable encryption are mentioned in table 2.1.

Table 2.1: Description of Different SE Schemes

Scheme Type	**Description**	**Merits**	**Demerits**
Symmetric SE (SSE) [17]	Data queried and retrieved via trap-door generation	High efficiency, good security and reliability	Prone to information leakage
Public Key Encryption with Keyword Search (PEKS) [18]	Data and indices encrypted with owner's public key	More secure than Symmetric searchable encryption (SSE)	Less efficient than SSE

Continuation of Table 2.1			
Scheme Type	**Description**	**Merits**	**Demerits**
Identity-Based Encryption (IBE) [19]	based on PKE wih key generated by user's identity	high efficiency, semantic security and provides access control	deterministic searching and queries patterns
Predicate encryption (PE) [20]	searching over encrypted data, queries based on tokens	no information leakage issues, more efficient than PEKS	requirement of high resources than PEKS
Inner Product Encryption (IPE) [21]	based on IBE schemes and payload hiding	efficiency and high security with access control	requirement of high resources
Multi-keyword Ranked Search Encryption (MRSE) [22]	based on searching algorithm with similarity of keywords	preservation of user's privacy	Limited keyword dictionary
Private Information Retrieval (PIR) [23]	data retrieval from CSP without any keyword or search queries pattern leakage	low resources, less cost of communications	does not support searching over plaintexts

Continuation of Table 2.1			
Scheme Type	**Description**	**Merits**	**Demerits**
Homomorphic Encryption (HE) [24]	enable computations and searching capability over encrypted data	end-to-end data privacy	high computational and storage overheads
End of Table			

2.3 Homomorphic Encryption (HE)

Homomorphic encryption schemes enable users to efficiently analyze and operate on data without decryption, saving resources and time [25]. This entails that operations that have been carried out over encrypted data will yield the same results as if done on plaintexts and encryption carried out afterward. Similarly, the results would be the same if encrypted data undergoes some processing, the underlying plaintext will retain all its original properties. The operations performed in these cases can be additive or multiplicative where M_1 and M_2 are two different plaintexts of messages [26].

- Additive Homomorphic encryption:

  $$\mathcal{E}(M_1 \oplus M_2) = \mathcal{E}(M_1) \otimes \mathcal{E}(M_2)$$

- Multiplicative Homomorphic encryption:

  $$\mathcal{E}(M_1 \otimes M_2) = \mathcal{E}(M_1) \otimes \mathcal{E}(M_2)$$

Various homomorphic encryption techniques *i.e.* partial, full and somewhat homomorphic encryption are addressed in [27] [28]. Homomorphic encryption techniques are employed in cloud computing as it allows multiple operations on already encrypted data without disturbing the integrity of plaintexts. However, the drawback of homomorphic encryption is its slow computation time [29]. There are three types of homomorphic encryption:

1. **Fully Homomorphic Encryption (FHE):** FHE is the most powerful type of homomorphic encryption, allowing for arbitrary computations to be performed on ciphertext data without the need for decryption. FHE is computationally intensive and currently impractical for most use cases due to its high computational and memory requirements.

2. **Partially Homomorphic Encryption (PHE):** PHE allows for a single type of operation to be performed on ciphertext data without requiring decryption. This type of homomorphic encryption is typically faster than FHE but is still limited in its functionality.

3. **Somewhat Homomorphic Encryption (SHE):** SHE is an intermediate type of homomorphic encryption that can perform a limited set of operations on ciphertext data without the need for decryption. SHE can be faster and more efficient than FHE or PHE, making it a popular choice for practical use cases.

In general, the choice of homomorphic encryption type depends on the specific use case and the required level of functionality, performance, and security. A

comparison is given in table 2.2.

Table 2.2: Comparison of Different HE Schemes

Particulars	PHE	SHE	FHE
Number of Computations	1	1 (upto some complexity)	multiple
Operations Allowed	either addition or multiplication	either addition or multiplication	both addition and multiplication
Arbitrary Computations	not allowed	not allowed	allowed
Computation on Encrypted Data	yes	yes	yes
Limitations	one operation	limited circuit depth	huge memory requirement
Example	RSA	Gentry	Fujitsu

2.4 Preliminaries

2.4.1 Advanced Encryption Standard (AES)

AES is a block cipher [30] based on an iterative structure and Substitution-Permutation Network with a specified block length of 128 bits (16 bytes / 4 words). It implies that it processes a data block of 4 columns of 4 bytes (state) taking 128 bits of input i.e. plain-text along with key and outputs an encrypted block i.e. cipher-text. Since AES is a symmetric key algorithm, the same key is used for encryption as well as the decryption process. The key size, however, is flexible as

it can be 128 bit, 192 bits or 256 bits long.

$$CT = E_k(PT)$$

$$PT = D_k(CT)$$

$$where: PT = plain - text;$$

$$CT = cipher - text;$$

$$E = Encryption\ function$$

$$D = Decryption\ function$$

$$k = symmetric\ key$$

It is extremely difficult to launch attacks on AES and brute forcing an AES algorithm requires $2^{key-length}$ which renders the attempt ineffective and highly extensive. So far, AES is the most secure encryption mechanism being employed all over the research domain [31].

2.4.2 Paillier Homomorphic Cryptosystem

Pascal Paillier proposed the Paillier cryptosystem [32] based on probabilistic encryption. This partial homomorphic encryption scheme depicts additive homomorphic property and is IND-CPA secure. The basic structure of Paillier cryptosystem is shown in figure 2.1.

Paillier HE allows support for both addition as well as scalar multiplication such as:

1) **Key Pair Generation:**
   - Choose two large prime numbers p and q.
   - Compute $n = p * q$
   - Compute $\lambda = lcm(p-1, q-1)$
   - Choose g where $g \in Z_n^2$ and $gcd(g, n) = 1$.
   - A key pair; $p_k = (n, g)$ and $s_k = \lambda$.
2) **Encryption:**
   - Select message m, where $0 \leq m \leq n$
   - Choose $r \in Z_n^2$ and $gcd\ (r, n) = 1$.
   - Compute $c = g^m\ *\ r^n\ mod\ n^2$
3) **Decryption:**
   - A function L such that $L(x) = (x-1)\ /\ n$
   - Compute $L(c^\lambda\ mod\ n^2)\ /\ L(g^\lambda\ mod\ n^2)$
   - Compute $\mu = L(g^\lambda\ mod\ n^2)^{-1}\ mod\ n$
   - $m = L(c^\lambda\ mod\ n^2)\ /\ L(g^\lambda\ mod\ n^2)\ *\mu\ mod\ n$

Figure 2.1: Paillier Homomorphic Cryptosystem

- **Homomorphic Addition:** For two plaintexts messages m_1 and m_2 with corresponding ciphertexts *i.e.,* c_1 and c_2; the homomorphic addition is carried out by computing $(c_1\ *\ c_2)\ mod n^2 = g^{(m_1+m_2)\,*\,(r_1\,+\,r_2)}\ mod\ n^2$, where r_1 and r_2 are some random numbers used in their encryption.

- **Homomorphic Scalar Multiplication:** For a message m with a ciphertext c and a scalar number k, the homomorphic scalar multiplication is carried out as: $c^k\ mod\ n^2 = g^{mk}\ *\ r^{kn}\ mod\ n^2$ where r is a random number used in its encryption.

2.4.3 Image Hash

An image hash is a compact representation of an image, usually a string of numbers or characters that represent its visual characteristics [33]. Once the hash is

generated, the image can be compared to other images or identified in a database based on the hash.

Image ⟶ **Image Hash Function** ⟶ *hash of image* ⟶ **Hash Database**

Images are hashed by converting them into a standard format, reducing their size, and then using an algorithm to generate unique values based on their content. There are a variety of applications for image hashing, including image recognition, content-based image retrieval, and image verification [34]. Applications of image hashing include multimedia retrieval, social network analysis, and information retrieval [35].

2.5 Related Work

2.5.1 Object Detection Algorithms

Early research on pattern recognition laid the groundwork for modern object detection and computer vision in the late 1970s to 1980s. Several object detection algorithms have been put forward by researchers since then [36] [37]. The first algorithm to combine convolution neural networks (CNN) with the RP method was R-CNN [38]. It was dual-stage and required high running time and storage space while carrying out feature extraction of the region of interest (ROI). An enhanced version, fast RCNN [39] was presented by the use of SPPNET [40]. Another scheme FPN [41] is based on fast RCNN with the capability to process different image scales. A more improved dual-stage version; faster RCNN [42] was

proposed with better performance metrics but was also limited to slow real-time object detection as well as its latency in the training of customized data. Mask RCNN [43] is based on semantic segmentation masking with high accuracy but lacks efficiency in the detection of masks at the pixels' level.

YOLO [44] is a single-stage detection algorithm using Darknet with lesser requirement of storage and computational time. It is bound by its object proximity to accuracy issues. YOLO v2 [45] is based on the novel Dark Net19 with the capability to detect over 9000 object classes yet is inaccurate in detecting sizes of objects from blurry images. SSD [46] was proposed with combined features of RPN and YOLO. A single-stage algorithm RetinaNet [47] uses the function of focal loss and claims higher accuracy rates. R-FCN [48] is a dual-stage region-based object detection algorithm. YOLO v3 [49] is good for real-time object detection applications and supports Multiple Object Tracking (MOT) [50]. The purpose of MOT technology is to provide a continuous sequence of images/video frames, identify moving objects in each frame, and assign the same object an accurate and particular ID. Other MOT supporting algorithms include Simple Online and Real-time Tracking (SORT) [51], its enhanced version deepSORT [50], Joint Probabilistic Data Association Filter (JPDAF) [52], Multiple Hypothesis Tracking (MHT) [53] and YOLO v4 [54]. Object detection has been revamped as part of the development of YOLO v4. Its architecture is based on the CSPDarkent53. Spatial pooling is used in the backbone to improve receptiveness and identify the required properties of data images/video frames [54]. A timeline of various object detection algorithms is described in table 2.3.

Table 2.3: Description of Different Object Detection Algorithms

Year	**Name**	**Description**
2001	Viola - Jones Algorithm [55]	The Viola-Jones algorithm was one of the earliest object detection algorithms. It used Haar-like features and a boosting classifier to detect faces in images.
2005	Histogram of Oriented Gradients (HOG) [56]	The HOG algorithm used gradients of an image to detect object edges and was able to achieve high results in pedestrian detection.
2006	Deformable Part Model (DPM) [57]	The DPM algorithm introduced a part-based model for object detection, where an object is made up of several parts that can be detected independently.
2008	Deformable Parts Model (DPM) [58]	It was introduced with improved accuracy of object detection by modeling the parts of the object and their spatial relationships.
2011	Region-based Convolutional Neural Networks (R-CNN) [38]	R-CNN was introduced using deep learning techniques.

Continuation of Table 2.3		
Year	**Name**	**Description**
2014	Fast R-CNN [39]	Improved version of R-CNN was introduced, which significantly improved the speed and accuracy of object detection using a single deep neural network.
2015	Faster R-CNN [42]	The Faster R-CNN algorithm used a Region Proposal Network (RPN) to generate region proposals and a Fast R-CNN network to classify and refine them.
2015	FaceNet [59]	FaceNet used a convolutional neural network (CNN) for facial feature extraction.
2016	Single Shot MultiBox Detector (SSD) [46]	The SSD algorithm used a single network to perform both region proposal and object detection, achieving real-time performance.
2016	Multi-Task Cascaded Convolutional Neural Networks (MTCNN) [60]	Multi-Task Cascaded Convolutional Neural Networks (MTCNN) used a cascaded convolutional neural network to detect faces and extract facial landmarks.

Continuation of Table 2.3		
Year	**Name**	**Description**
2016	You Only Look Once (YOLO) [44]	The YOLO algorithm used a single neural network to predict bounding boxes and class probabilities straight from full images, also achieving real-time performance.
2017	Mask R-CNN [43]	The Mask R-CNN algorithm extended Faster R-CNN by adding a branch for predicting object masks in addition to bounding boxes and class probabilities.
2017	RetinaNet [47]	The RetinaNet algorithm used a focal loss function to address the class imbalance problem in object detection, achieving state-of-the-art results.
2018	YOLO version 2 [45]	The YOLO v2 achieved better accuracy in the detection of objects in higher resolution images as well as lightweight architecture.
2018	YOLO version 3 [49]	The algorithm boasted a higher accuracy and better performance in case of detection of smaller objects.
2018	Cascade R-CNN [61]	The Cascade R-CNN algorithm used a cascade of classifiers to refine object detections, achieving state-of-the-art accuracy on some benchmarks.

Continuation of Table 2.3		
Year	**Name**	**Description**
2019	EfficientDet [62]	The EfficientDet algorithm used an efficient backbone network and compound scaling to achieve state-of-the-art accuracy and efficiency on object detection benchmarks. It has several versions, including EfficientDet-D0 to EfficientDet-D6.
2019	CenterNet [63]	The CenterNet algorithm used a keypoint estimation approach to detect object centers and their corresponding bounding boxes. It has several versions, including CenterNet-104, CenterNet-Hourglass-104, and CenterNet-MobileNetV2.
2019	FreeAnchor [64]	The FreeAnchor algorithm used a novel anchor-free approach to object detection, achieving state-of-the-art accuracy on some benchmarks.
2019	FSAF (Feature Selective Anchor-Free) [65]	FSAF was introduced, which is an anchor-free object detection model that uses a novel feature selection strategy to identify object locations in feature maps without explicit anchor boxes.

Continuation of Table 2.3		
Year	**Name**	**Description**
2020	YOLO version 4 [54]	YOLO v4 used a number of innovations to improve accuracy and speed, including a larger network, multi-scale training, and a new data augmentation strategy.
2020	EfficientPS [66]	The EfficientPS algorithm used a novel feature pyramid architecture, efficiently improving object detection accuracy.
2020	SpineNet [67]	The SpineNet algorithm used a novel architecture for backbone networks in object detection, achieving state-of-the-art accuracy and efficiency on some benchmarks.
2020	RepPoints [68]	The RepPoints algorithm used a set of representative points to represent objects, which were then used for object detection/segmentation.
2020	DEtection TRansformer (DETR) [69]	The DEtection TRansformer (DETR) algorithm introduced a transformer-based architecture for object detection, which directly predicted object instances and their positions without requiring region proposals.

Continuation of Table 2.3		
Year	**Name**	**Description**
2021	Sparse R-CNN [70]	The Sparse R-CNN algorithm used a sparsity-inducing regularization to learn a sparse set of region proposals, achieving state-of-the-art accuracy while reducing computation cost.
2021	Deformable DEtection TRansformer (DETR) [71]	The Deformable DEtection TRansformer (DETR) algorithm used deformable transformers to improve the performance of DETR on small objects and densely packed scenes.
2021	YOLO version 5 [72]	Improved version in terms of video frames/webcam detection with easy weights transfer. More lightweight and fast than the previous versions.
2022	Panoptic FPN [73]	The Panoptic FPN algorithm introduced a unified architecture for both instance segmentation and semantic segmentation, achieving state-of-the-art results on several benchmarks.
2022	YOLO version 6 [74]	Improved version in terms of small object detection but low flexibility and stability as compared to YOLO v5.

Continuation of Table 2.3		
Year	**Name**	**Description**
2022	YOLO version 7 [75]	YOLO v7 introduced an object detection model with higher accuracy than previous versions but low speed than v6.
End of Table		

Comparing deep learning image/object detection algorithms can be a complex task as there are many different factors to consider. However, some key areas of comparison include accuracy, speed, and efficiency.

- **Accuracy:** The accuracy of an image detection algorithm is determined by its ability to correctly identify objects in an image. One way to measure accuracy is through the use of the mean average precision (mAP) metric. A higher mAP score indicates better accuracy.
- **Speed:** The speed of an image detection algorithm is determined by how quickly it can process an image and output the results. This can be measured in terms of frames per second (FPS) or inference time.
- **Efficiency:** The efficiency of an image detection algorithm is determined by how well it performs while using the least amount of resources possible, such as memory and processing power.

In summary, the choice of deep learning image detection algorithm depends on the specific needs of the application. If accuracy is the top priority, Faster R-CNN or

RetinaNet may be the best options, while YOLO or SSD may be preferred if speed is the most important factor. Additionally, it's worth considering the efficiency of the algorithm to ensure it runs well on the available hardware resources.

2.5.2 Literature Review

Over the years, research into image processing techniques has resulted in a need for high computations and resources. There are various image processing techniques available providing a vast field for image searching mechanisms [76] [77]. This literature review aims to provide an overview of the current state-of-the-art image-based SE schemes for cloud storage and retrieval [78] [79].

Image-based searchable encryption (SE) schemes for the cloud have become increasingly popular in recent years due to the growing need for secure and efficient cloud storage of image data. In these schemes, image data is encrypted and stored in the cloud, while users can search for specific images using keywords or image content without compromising the security of the data [80] [81].

The first approach of SE over image data is credited to [82] via Scale Invariant Feature Transform (SIFT) [83]. A major drawback of the research was that it lacked privacy preservation property and had a huge overhead on the user's end. By using a multi-cloud model, [84] addresses these vulnerabilities by preserving the user's privacy while preserving the original SIFT features of the image. This protocol involved two cloud servers and operations involved the division and extraction of image features. This scheme was limited by its deterministic approach

as query patterns and no randomness factor and was thus prone to traceability attacks.

Using Paillier cryptography, a scheme for extracting image features was proposed in [85]. The scheme privacy preservation using SIFT (PPSIFT) discussed a comprehensive security analysis based on discrete logarithm problem (DLP) and RSA encryption algorithm and showed that it provides various attributes of image feature detection *e.g.* descriptor matching, local extrema extraction, and descriptor calculation over an encrypted domain with a single pre-communication round for data synchronization. Authors claimed that their proposed scheme provides privacy preservation using SIFT and is secure against ciphertext-only attack (COA). This scheme, while providing more security than simple homomorphic-based SIFT is however computationally extensive and requires a high-powered server.

The authors in [86] analyzed the design goals and technical challenges associated with a cloud-based image processing system. Design targets as set by this research are three-fold with the first target as the functionality *i.e.* selection of a specific algorithm from various image processing techniques available according to the available resources. The second target is set as security requirements with regard to safeguarding the contents of data (images) from any modification or theft etc. The efficiency of the overall scheme is kept as the third and final design target which includes the operational, computational as well as communication complexity. The research incorporated both local and global feature-based image searches. Global feature implies the search over the entire image. It involves a single round of queries and responses between the user and the cloud server. It

was carried out by RGB histogram [87] after which the following color descriptor was computed *i.e.* Colour Layout Descriptor (CLD) and Colour Structure Descriptor (CSD). Local feature extraction for image search means the search over patches or small vectors containing image data. It involves two rounds of queries and responses between the cloud server and user/client. SIFT or Histogram of oriented gradients (HOG) was used for feature extraction from image data.

However, it was seen that the ciphertexts release considerable information leakage about the plaintexts and simple SIFT or HOG algorithms are insufficient to protect the contents of images from the cloud server. Similar was the case in global feature-based image searches. Thus, an additional encryption scheme of homomorphic encryption is studied and applied to gain sufficient security over images before outsourcing data to cloud storage. The overall efficiency of the system is based on 3 outlooks *i.e.* computational efficiency at the user's and cloud's end and communication overhead between cloud and user. However, it poses an issue regarding the efficiency of the system as additional encryption mechanisms imply an increase in complexity and security but a decrease in overall efficiency. One of the solutions to this problem is to create a multi-server environment. Another solution is to employ a somewhat homomorphic encryption (SHE) scheme including secure extraction and detection of image features as well as secure retrieval of images and matching mechanisms. The drawback with SHE is its finite operational capabilities with a limited number of multiplications. It can be resolved by a combination of SHE with any technique of multiparty computation. It will provide better efficiency and better performance than the two schemes being employed

separately.

Using somewhat homomorphic encryption (SHE), a Hahn Moment-based approach preserves the privacy and confidentiality of reconstructed images [88]. The merit of this research was its low utilization of computational power and resources as compared to other image processing techniques *i.e.* Discrete Wavelength Transform (DWT) and Discrete Cosine Transform (DCT) [89]. Another merit of this technique is the usage of Hahn's calculations and equations with their simple additive and multiplicative operations, lack of or very small high-end computations, low noise sensitivity, and ability of good quality restoration especially in this particular case of orthogonal Hahn's equations [90]. Its major application lies in the domain of pattern and image assessment as well as its identification [91]. However, the limitation of this model, though based on privacy-preserving homomorphic encryption (PPHE), is its deterministic approach to image searches. It means that search queries for an image will always yield the same results which can lead to sniffing attacks.

Another approach to carrying out searching for an image feature similarity in a cloud setting was presented in [92]. It addressed extraction of both local and global features using Earth Mover's distance measure, as well as searchable index generation. The data owner creates a data set of features (both global and local) and generates an index for efficient searching. Both these images and indices are then encrypted at the owner's end and outsourced to the cloud. The user while searching has to create an encrypted query through the extraction of a feature vector from the image and send it to CSP which will respond back with images

after comparison with the similarity index. The user has to decrypt the image at its end. This scheme can be applied for Content-based Image Retrieval (CBIR) but it will require huge storage requirements. The scheme has the demerit of its increasing computational complexity with an increase in the number of data images being stored on CSP. More storage will require more searching time and thus will bring down the overall efficiency.

In [93], a technique for image data privacy preservation based on Linear Binary Pattern (LBP) was proposed to extract features from images and converting images to matrices after their encryption. To safeguard the image content, the central value of the image-turned-matrix is set as a specific binary value. The user encrypts the image's MSB by XOR operation with the Bit plane randomization method before outsourcing it to the cloud where true random numbers are employed to ensure the confidentiality and privacy of images. Upon any query from a user, CSP performs its operations on encrypted data. The merit of this technique is its low operational overhead at the cloud's end and no or little additional communication issues between the user and the cloud server. This scheme is limited as it is only verified over grayscale image data and is feasible for feature-based search. It does not contribute towards content-based image retrieval (CBIR). Another major drawback includes the lack of security for search queries from the user's end and its susceptibility to threats regarding insecure search patterns.

A novel scheme was presented in [94] that took elements from Secure Modular Hashing (SMH) [95] and K-means [96]. Secure modular hashing is a technique of determining the distance between 2 signals using hashing approach on a modular

embedded system. K-means is used for the estimation of similarity among a cluster between 2 or more vectors. The similarity forms an inverse relationship with the vector distance *i.e.* maximum the distance, lower the similarity. The authors in [94] used K-means to generate an index table. Images are encrypted after the extraction of feature vector(s) at the owner's end. These encrypted images and indices are then stored over the cloud. Data user requests CSP with encrypted feature vector as a search query. The CSP responds with the resulting image after correlating the similarity between the received feature vector and the already stored index. The user will decrypt the received image at its end. The key(s) used for encrypting feature vector(s) and received image(s) are preshared by the owner. This scheme is limited due to its deterministic approach toward their search queries. As a result, their system is insecure, as search patterns constitute a critical component of user privacy.

In [97], a technique called EPCBIR (Efficient and Privacy-Preserving CBIR) that utilizes cloud assistance was proposed utilizing kNN and LSH. Although their approach offers a ranking-based image search mechanism, it requires significant computational resources. The article [98] presents a privacy-preserving system called PUPPIES that leverages a dynamic partial image-sharing approach. This method allows data owners to define specific private regions inside an image, such as a face or a social security number, and assign different privacy policies to each user.

The scheme in [99] describes a secure recovery mechanism for encrypted YUV color photographs using DCT. The resulting DC coefficients and AC coefficients,

as well as the other two color components, are encrypted using stream cipher technology before outsourcing to the cloud. The Manhattan distance of their respective histograms is calculated to determine the similarity between the query trapdoor and the database picture. Finally, the encrypted photos that are the most similar to the query image are retrieved by the user(s).

The authors proposed SEISA in [100], an approach that allows for secure and efficient image search while maintaining access control. The method employs an encrypted image index to achieve efficient searches while keeping image content private. Furthermore, access control is implemented by requiring users to provide a valid key or policy in order to decode the index and view the photos. Experiment results showed that the proposed strategy is successful and efficient. Overall, the SEISA technique provides a feasible solution for secure image search that includes access control.

In the scheme PIC [101], a privacy-preserving CBIR scheme was presented for large-scale data in the cloud. This approach, called PIC, permits encrypted image searches with efficient access controls determined by data owners. Meanwhile, [102] discusses encrypted image searching in the mobile cloud domain. Additionally, [103] introduced a privacy-preserving image search (PPIS) utilizing a convolutional neural network (CNN) for large-scale medical image data. The authors claim that the PPIS approach provides secure search queries and privacy preservation of image data.

In [104], Li *et al.* presented a new scheme for multi-user cloud-connected image data that employed CNN, bilinear mapping, and proxy re-encryption. Meanwhile,

in [105], Y. Duan *et al.* proposed a CNN-based retrieval scheme for medical image data that utilized Euclidean distance and kNN. Lastly, Wang *et al.* introduced a framework [106] for secure image retrieval in a multi-owner multi-user environment without the need for key sharing.

The authors developed a unique technique in [107] using CNN for content-based image retrieval in a cloud environment with a multi-share creation scheme. The method secures the storage and retrieval of critical photos by dividing them into numerous shares and encrypting each share with a unique key. Image retrieval is accomplished by reconstructing the original image from encrypted shares. Deep learning techniques are used in this scheme to improve the accuracy of picture retrieval results.

An approach is presented for efficient indexing and retrieving medical images from a cloud-based system using hashing methods [108]. The solution overcomes privacy issues by offloading the retrieval procedure to the cloud while maintaining image security. When compared to existing state-of-the-art strategies, the suggested method achieves better outcomes in terms of retrieval speed.

Authors in [109] proposed a technique to perform privacy-protected image retrieval. The scheme preserves the privacy of the images by encrypting them before being indexed and stored in the cloud-based system. The retrieval process is then performed on the encrypted images without revealing any sensitive information.

A novel approach for learning image attributes and generating compact binary codes for indexing and retrieving images is proposed [110]. The anonymity of

images is protected by encrypting the binary codes before sending them to the cloud-based system using a homomorphic encryption algorithm. The proposed scheme claims better retrieval accuracy while maintaining image privacy than many state-of-the-art schemes. The authors in [111] discussed methods for detecting similar images, with a focus on using homomorphic encryption and secure multiparty computation techniques to achieve security and privacy preservation.

Sultan *et al.* proposed a novel homomorphic-based image retrieval scheme in [112] for cloud-connected devices. The scheme carried out the encryption of images at pixel levels and was based on Paillier cryptosystem for encrypted image searching. It used probabilistic trapdoors for image queries and claims to ensure search pattern security. A novel approach for similarity searching and retrieval for images of *jpeg* format is presented in [113]. The scheme is based on Bag-of-Words (BoW) Model and local Markov Feature [114]. The search for similar images involves extracting local Markov features directly from the encrypted image file and utilizing the BoW model to exploit these features, achieving good retrieval accuracy. The server is responsible for outsourcing image storage, feature extraction, and image searching, thereby reducing the burden of the data owner.

Li *et al.* proposed a deep learning-based privacy-preserving scheme for JPEG image retrieval [115]. By utilizing deep learning techniques, the scheme aims to protect the privacy of image data during retrieval by extracting and comparing image features without revealing their original content. It preserves the confidentiality of the original data while retrieving similar images efficiently and accurately. This scheme has practical applications in a number of domains, such as medical imaging

and law enforcement, where the privacy of sensitive image data is paramount.

2.6 Security Challenges

Similar image retrieval schemes for cloud-connected devices face several security challenges, some of which are as follows:

- **Confidentiality:** The images over the cloud may contain sensitive information, such as personal or corporate data, and need to be protected from unauthorized access or disclosure.
- **Data integrity:** The images stored on the cloud must remain unaltered, and their integrity must be protected from malicious attacks or unauthorized modifications.
- **Authentication:** It's essential to ensure that only users that have authorized access can retrieve the images stored on the cloud. Authentication mechanisms need to be put in place to prevent unauthorized access.
- **Search Pattern Security:** It is important to ensure search pattern security such that no information is accessible to any adversary if queries and their outcomes are stored to gather patterns.
- **Data loss or theft:** The images stored on the cloud may be subject to data loss or theft, leading to the compromise of sensitive information.

To address these security challenges, cloud-based image retrieval systems must im-

plement robust security measures such as access control, encryption, data backup, and recovery mechanisms, and implement security monitoring and logging capabilities to detect and respond to security incidents promptly.

2.7 Design Goals

A searchable encryption scheme for cloud storage should be designed with several goals in mind to ensure that it provides both security and functionality. Some design goals for a searchable encryption scheme for cloud storage include:

1. **Efficient Searchability:** The scheme should allow for efficient searching and retrieval of encrypted data from the cloud, without requiring the data to be decrypted first. The scheme can be applied to large-scale datasets and does not significantly impact the performance of the cloud system.

2. **Security:** The scheme should be designed with strong security guarantees, such as resistance to known attacks, including dictionary and statistical attacks, and provide proof of security.

3. **Privacy Preservation:** The scheme should ensure the privacy of data stored on the cloud by encrypting it in a way that prevents unauthorized access or decryption by third parties, including the cloud service provider.

4. **Scalability:** The scheme should be scalable to support large amounts of data and users while maintaining efficient searching and retrieval times.

5. **Usability:** The scheme should be user-friendly, easy to implement and maintain, and integrate well with existing cloud storage and searching infrastructure.

Overall, the design goals for a searchable encryption scheme for cloud storage should aim to balance the competing requirements of security as well as privacy preservation, efficient searching, scalability, and usability to provide a practical and effective solution for securing data in the cloud. The trade-off between security attribute and performance requirements varies with different use cases.

2.8 Summary

The chapter discussed Searchable Encryption (SE) in detail along with its different types including Homomorphic encryption. It also presented some preliminaries. A comprehensive analysis of object detection schemes was presented. A literature review of existing SE schemes for images stored at a Cloud server was carried out in detail and a comparison was drawn over some common metrics. Lastly, some challenges for image retrieval were put forward and design goals for our research were presented. Chapter 3 will present the detailed system model.

Chapter 3

System Model

3.1 Overview

This chapter puts forward the system model with a detailed threat model, security goals, and assumptions made in this research. The threat model provides the security challenges posed to the proposed scheme with the adversary's capabilities.

3.2 System Model

3.2.1 Network Model

The network model compromises surveillance cameras/sensors, data owner/user, and cloud server CS. The network model is shown in figure 3.1. The video feed is generated by the smart cameras with facial recognition and is received by the owner. The owner extracts the facial images, calculates hash values and encrypts them along with image data, and outsources them over to the cloud server. The

cloud server is the major storage entity over which the image search is carried out. The owner or any authorized user can request any image(s) similar to the images stored in the cloud. For searching, the user generates a trapdoor by encrypting the hash value of the query image. After the trapdoor is generated and sent over to the cloud; the cloud conducts a similarity search over facial images and then sends the encrypted results back to the user. The user then decrypts the results, checks for mapped *IDs* of the images with the faces, and makes a request for specific images containing those required people. The cloud retrieves the images and sends them over to the user which decrypts the images at its end.

The encryption keys (including AES symmetric key and Paillier's public/private key pair) are generated at the data owner's end. The facial detection is carried out by MTCNN [60]. The images are encrypted by AES encryption and facial image hash values are encrypted by Paillier HE. The query image hash is encrypted by Paillier's public key. The search is carried out over the cloud server and retrieved images are then decrypted by the user via AES decryption.

3.2.2 Threat Model

In the proposed scenario, the threat model consists of two entities: the data owner/user and the cloud, with images being the stored data. The primary objective of an adversary $\mathcal{A}$ is to gain access to the image data stored on the cloud. Due to all communication occurring over a public channel between the owner and the cloud, interception and subsequent attacks to expose the underlying data are easily achievable by the adversary. The adversary could be either an outsider or the

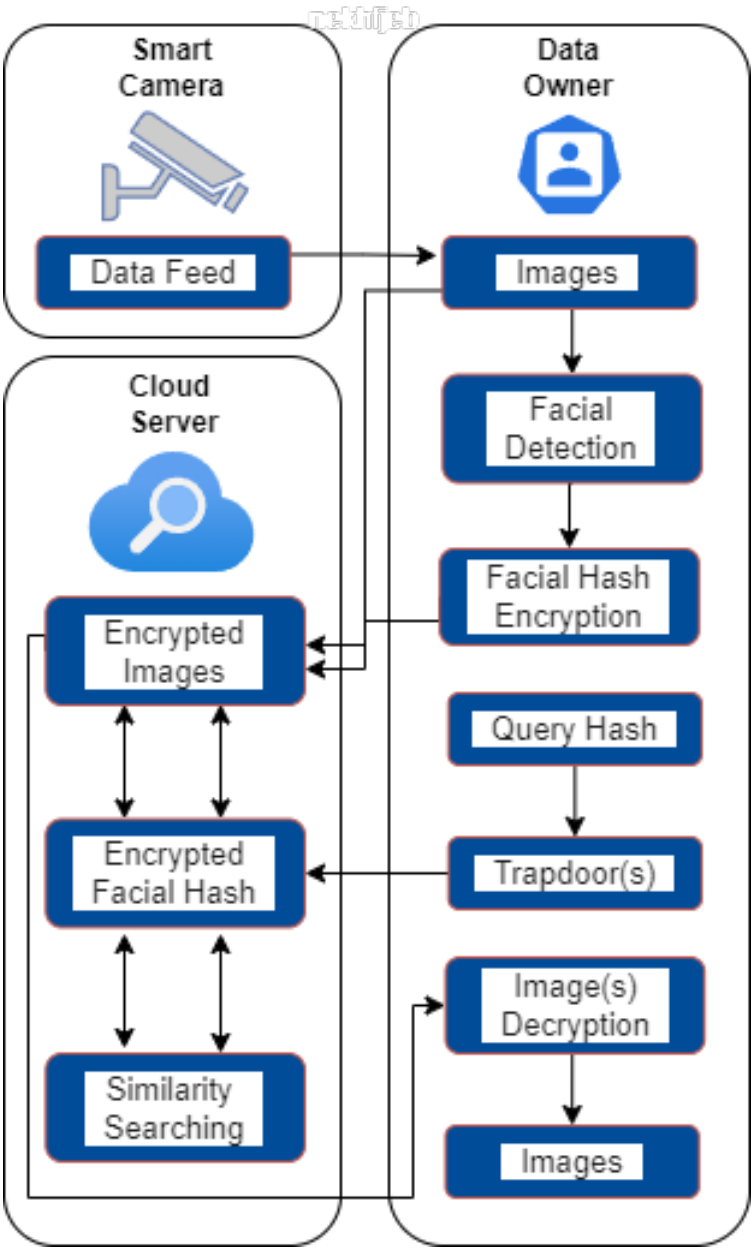


Figure 3.1: Network Model

cloud itself, possessing the capabilities or conditions listed below:

- Only passive attacks can be launched by the cloud server so it can monitor the network traffic and capture information about the network activity.
- The adversary is constrained by the storage as well as computational resources and the amount of time available to be able to gain access to images

in polynomial time.

- The adversary has the ability to trace previous search inquiries, search outcomes, and communication patterns between the owner/user and CS, and can exploit the information gathered, to their benefit.

3.2.3 Security Assumptions

The following security assumptions have been made in this research:

- The owner is presumed to be completely trustworthy and to pose no harm to the system's security.
- The video feed/ image frames generated by the camera are received by the owner over a secure channel.
- The cloud server is deemed *trusted but curious* implying that the cloud may try to get any available information about the data stored over it.

3.2.4 Security Goals

The security goals that are targeted for this research include:

- **Adaptive Security:** The approach should be secure in the known-ciphertext model. In an adaptive adversarial model, this means that the CS should not be able to gather any information about the search, even if they are familiar with past queries made.

- **Authentic Trapdoor Generation:** The trapdoor can be generated only with the necessary keys by the authorized user(s).

- **Search Pattern Security:** Search pattern security in the cloud refers to measures taken to protect the secrecy and privacy of search queries and outcomes performed by users on data stored in the cloud. This involves safeguarding against unauthorized access, interception, or manipulation of search queries and results by adversaries who may attempt to exploit vulnerabilities in the cloud infrastructure or network.

- **Trapdoor Indistinguishability:** The property of trapdoor indistinguishability is used to protect data stored in the cloud by allowing users to generate trapdoors that are indistinguishable from one another. It implies that even if an adversary gains access to more than one trapdoor, it should be nearly impossible to distinguish these from each other and gain any meaningful insight into the underlying data.

3.3 Summary

In this chapter, the system model was discussed with a detailed threat model, security goals, and assumptions made in this research. Chapter 4 will present the proposed scheme for secure similar image retrieval from Cloud.

Chapter 4

Proposed Work

4.1 Overview

This chapter puts forward the proposed scheme for homomorphic-based similar image retrieval for cloud-connected devices. The system model is discussed with a detailed threat model, security goals, and assumptions made in this research. All phases of the proposed scheme are presented in detail along with their algorithms. The research includes the following contributions:

- A comprehensive literature review and analysis of object detection algorithms and existing privacy-preserving image-based searchable encryption techniques for cloud-connected devices.
- A novel scheme for similar image searching based on homomorphic encryption is proposed for cloud-connected devices. The scheme is based on probabilistic trapdoors and ensures the privacy preservation of image data.

- Implementation and testing of the proposed scheme are carried out over a real-world data set in order to assess its security and performance in terms of complexity, computation, and storage overheads.

The security analysis is carried out in detail in chapter 5 and the performance analysis is presented in chapter 6 respectively.

4.2 Proposed Scheme

The proposed similar-image searching scheme has a two-step process where the faces in the images are detected by MTCNN [60]. Next, the detected image is encrypted using Advanced Encryption Standard (AES) encryption. To improve the efficiency and performance of the algorithm and reduce storage and computation overhead, the proposed scheme uses two different encryption techniques. In addition to AES encryption, the algorithm uses Paillier homomorphic encryption to provide secure searching via trapdoors over encrypted data. The faces detected are then passed through a perceptual hash function whose resultant these values are then encrypted using the Paillier HE. These encrypted image hashes are used to carry out similar image searches, ensuring the security and confidentiality of the data. The proposed scheme consists of six major phases that are discussed in detail as follows:

4.2.1 Key Generation Phase

The Paillier cryptosystem uses a probabilistic algorithm for key pair generation that generates a Public Key and a Secret (Private) Key. This algorithm requires an input parameter, "pb," which determines the number of bits used to generate a prime number. The output of the algorithm is a secret (private) key, denoted by s_k, and a public key, denoted by p_k. To generate these keys, the algorithm randomly generates two prime numbers, p and q, that are independent of each other, based on the "pb" parameter. Using these prime numbers, the Secret Key s_k and Public Key p_k are then computed. The Secret Key s_k is kept secret and is only used for decryption, while the Public Key p_k is used for encryption. The secret key s_k is also used as AES symmetric key. The algorithm for the key generation phase is shown in figure 4.1.

Algorithm 1 Key Generation $(s_k, p_k) \leftarrow KG(pb)$

Generate p = getprime (pb,RN)
Generate q = getprime (pb,RN)
Let $n = p * q$
while $g = RN();\ GCD(g, n^2) \neq 1$ **do**
    Compute λ = LCM (p-1,q-1)
    Compute Modular Multiplicative Inverse:
    $\mu = (L(g^{\lambda}) mod n^2)^{-1} mod\ n$
    Compute: $l = (pow(g, \lambda, n^2) - 1)/n$
    Calculate: $gmu = libnum.invmod\ (l, n)$
end
return $s_k = (\lambda, \mu)$, $p_k = (n, g)$

Figure 4.1: Key Generation Phase

4.2.2 Image Encryption Phase

All the image data I_i is encrypted by AES using the AES symmetric key s_k to get the encrypted images E_{I_i}. The algorithm for image encryption is shown in figure 4.2.

Algorithm 2 Image Encryption $E_{I_i} \leftarrow E_s(I_i, s_k)$

for $i \leftarrow 0$ **to** *M; M are number of images* **do**
  $E_s(I_i, s_k) = E_{I_i}$
end
return E_{I_i}

Figure 4.2: Image Encryption Phase

4.2.3 Facial Hash Encryption Phase

In the facial hash encryption phase, the facial images are passed through an image hash function to obtain hash values F_{h_i} which are then encrypted via the Paillier cryptosystem using a public key p_k in a loop that utilizes the "$p()$" function. This function is an exponential function that raises the input parameters, namely g, H_{F_i}, n^2, to generate the encrypted facial hashes $E_{H_{F_i}}$. The algorithm is shown in figure 4.3.

4.2.4 Trapdoor Generation Phase

The user selects a query image, and calculates its hash value H_{Q_I}. The hash value is then encrypted by the Paillier homomorphic encryption using the public key p_k, similar to the facial hash encryption phase. The trapdoor T_Q is generated as a result as shown in figure 4.4.

Algorithm 3 Facial Hash Encryption $E_{H_{F_i}} \leftarrow Enc(p_k, H_{F_i})$

for $i \leftarrow 0$ **to** N; *N are the number of facial images* **do**
  for $i \leftarrow 0$ **to** F_i **do**
    $H_{F_i} = h\ (F_i)$
    Enc $(g^{H_{F_i}}.r^n)\ \%\ n^2$
    $E_{H_{F_i}} = p\ (g, H_{F_i}, n^2)$
    $E_{H_{F_i}}$ = write (H_{F_i})
  end
end
return $E_{H_{F_i}}$

Figure 4.3: Facial Hash Encryption Phase

Algorithm 4 Trapdoor Generation $T_Q \leftarrow TrG(p_k, H_{Q_I})$

for *all query image* Q_I **do**
  $H_{Q_I} = h\ (Q_I)$ Enc $(g^{H_{Q_I}}.r^n)\ \%\ n^2$
  $E'_{H_{Q_I}} = p(g, H_{Q_I}, n^2)$
  T_Q = write $(E'_{H_{Q_I}})$
end
return T_Q

Figure 4.4: Trapdoor Generation Phase

4.2.5 Similar Searching Phase

In this phase, the actual search for a similar image takes place at the cloud server. The trapdoor that is generated in the previous phase T_Q is multiplied by a scalar value of "-1" following the property of scalar multiplication of Paillier cryptosystem [32]. The result of such multiplication is then added with all the existing encrypted hash values E_{F_i} and the resultant values are stored in a vector V_R. The vector V_R is then returned to the user. The user decrypts it by using secret key s_k, checks for similar image results according to the predefined similarity threshold. The user then makes a specific request for image(s) by providing its ID in relation to the mapping of facial image hashes. The algorithm is shown in figure 4.5.

Algorithm 5 Similar Search $E_{I_i} \leftarrow SimSrch(E_{F_i}, T_Q)$

for $i \leftarrow 0$ **to** E_{F_i} **do**
  $T'_Q = (-1)\ T_Q$
  A = $A_{DD}(T'_Q, E_{F_i})$;
  $V_R = \sum_{a=1}^{i}(A)$**end**
  return V_R
  At User's End:
  for $y \leftarrow 0$ **to** V_R **do**
    if Dec $(V_{R_y}, s_k).get(V_{R_y}) \geq V_{th}$
    $E_{I_i} = getImage(V_{R_y})$
  end
  return E_{I_i}

Figure 4.5: Similar Searching Phase

4.2.6 Image Decryption

All the retrieved image data E_{I_i} is decrypted by AES at the user's end using the AES symmetric key s_k to get the images I_i. The algorithm for image decryption is shown in figure 4.6.

Algorithm 6 Image Decryption $I_i \leftarrow D_s(E_{I_i}, s_k)$

for $i \leftarrow 0$ **to** *N; N are number of encrypted images* **do**
  $D_s\ (E_{I_i},\ s_k) = I_i$
end
return I_i

Figure 4.6: Image Decryption Phase

4.3 Addition / Deletion of Images

The proposed scheme allows the data owner to add or delete single or multiple images at any instant in time. Since the proposed scheme follows a run-time probabilistic trapdoor-based searching algorithm, there is no maintenance of the

index tables or ranked entries. For all trapdoors that are generated to carry out searching, a one-to-many search sequence is carried out from scratch which will have no impact on the scheme if more images are added to the existing ones or if one or multiple are deleted from the stored image data at the cloud server.

4.4 Summary

This chapter presented the proposed scheme for homomorphic-based similar image retrieval for cloud-connected devices. All phases of the proposed scheme were presented in detail along with their algorithms. The properties of correctness and soundness were discussed as well. The security analysis is carried out in detail in chapter 5.

Chapter 5

Security Analysis

5.1 Overview

In this chapter, the security definitions are revisited in detail. The security analysis is presented in terms of the aforementioned definitions, leakages, soundness, and correctness of the proposed scheme. Lastly, a comparison among the latest schemes is drawn with respect to different security attributes.

5.2 Security Definitions

The research follows the security definitions introduced by Tahir *et al.* in [116]. The definitions are extensively employed and widely recognized in probabilistic trapdoor-based searchable encryption schemes, and are discussed as follows:

5.2.1 Adaptive Security

This implies the ability of an adversary $\mathcal{A}$ to make queries depending on the outcomes of earlier inquiries. A scheme is thus considered secure with respect to Adaptive Indistinguishability if the adversary $\mathcal{A}$ is unable to distinguish (provided with its record of earlier search queries) between two different data chunks constructed adaptively with the same length, similar trapdoors, and similar search patterns; with a probability outcome of greater than 1/2.

5.2.2 Non-Adaptive Security

The term non-adavtive security entails that an adversary $\mathcal{A}$ is not capable of making queries depending on the outcomes of earlier searches. A scheme is considered secure with respect to Non-adaptive Indistinguishability if the adversary $\mathcal{A}$ is unable to distinguish between two non-adaptively constructed different data chunks of same length, similar trapdoors and search patterns with a probability outcome of greater than 1/2.

5.2.3 Query - Trapdoor Indistinguishability

Query - Trapdoor indistinguishability is termed as a searching procedure performed through encrypted trapdoors generated by unencrypted queries. A random, probabilistic trapdoor is generated for each query, so a duplicate search for the same query results in two completely different trapdoors, and neither reveals information about the query made. The adversary $\mathcal{A}$ is not able to differentiate

between the two trapdoors, even if it maintains an adaptive query history and associated trapdoors. In order to predict contextually relevant query information, the $\mathcal{A}$ opponent must perform extensive operations as well as store large amounts of data in polynomial time.

5.2.4 Trapdoor - Image Indistinguishability

The level of complexity in a homomorphic-based SE protocol is closely tied to the Trapddor-Image indistinguishability. To ensure that the associated trapdoors remain secure during searches, the queries, trapdoors, and similar image searching should be sufficiently complex to prevent any leakage of information about the facial images. This means that, even in the event of an adaptive history (consisting of query, trapdoor, and facial images), the trapdoor must remain indistinguishable when the same search term is used again. Additionally, even minor changes to the query should result in significant alterations to the trapdoor, causing the search results to differ significantly from previous searches and vice versa. By making the trapdoor unpredictable, adversaries are unable to predict which image will be retrieved from the list of encrypted facial image hashes, thereby ensuring both the security of queries and the privacy of user(s).

5.2.5 Search Pattern Security

The scheme provides secure search patterns due to the generation of probabilistic trapdoors. Secure search pattern refers to the ability of an adversary $\mathcal{A}$ in deter-

mining if the same query is being made again. Because our technique is based on probabilistic trapdoors, the trapdoor for the same keyword will be different when created at various timestamps and thus efficiently conceals the search pattern.

5.2.6 Chosen Plaintext Attack (CPA) Security

The scheme is labeled as IND - CPA secure if an adversary $\mathcal{A}$ is unable to distinguish between the underlying chosen plaintext for encryption as a result of a security game with a probability of no higher than 1/2 *i.e.* if any probabilistic polynomial time adversary has just a small advantage over random guessing. Moreover, if security definitions of *Query - Trapdoor Indistinguishability* and *Trapdoor - Image Indistinguishability* are met, it implies that the scheme is IND-CPA secure as well.

The scheme is labeled as IND - CPA secure, if any probabilistic polynomial time adversary has just a small advantage over random guessing regarding output ciphertext. Let Setup ($\mathcal{KP}$, $\mathcal{E}$, $\mathcal{D}$) for scheme be defined for this game representing public/private key pair generation, encryption, and decryption respectively, the

game follows as:

$$(s_k, p_k) \leftarrow KG(pb)$$

$$E_{I_i} \leftarrow Enc_s(I_i, s_k)$$

$$E_{H_{F_i}} \leftarrow Enc(p_k, H_{F_i})$$

$$(m_0, m_1) \leftarrow \mathcal{A}(p_k)$$

$$a \leftarrow \{0,1\};\ m_a \leftarrow \mathcal{A}(x)$$

$$Output(\hat{a} \stackrel{?}{=} a)$$

$$if \hat{a} = a;\ output\ 1; otherwise\ output\ 0$$

The scheme is labeled as IND - CPA secure if an adversary $\mathcal{A}$ is unable to distinguish between the underlying chosen plaintext for encryption as a result of a security game with a probability of no higher than 1/2 *i.e.* if any probabilistic polynomial time adversary has just a small advantage over random guessing.

5.3 Security Analysis

This security analysis of the proposed scheme is presented with a game-based approach as follows:

5.3.1 Game 1: Query-Trapdoor Indistinguishability

Suppose there are multiple query images, denoted as Q_{I_1}, Q_{I_2}, ..., Q_{I_i}, for all the images I_i. The game involves three phases between an adversary and a challenger.

- *Query Phase:* The process begins with the challenger generating encrypted trapdoors for multiple facial images against the images I_i. Next, the adversary sends a query image Q_{I_i} to the challenger, who in turn responds with the corresponding encrypted trapdoor T_{Q_i}. This cycle of queries and responses continues to take place until the adversary has gathered a polynomial number of query-trapdoor pairs.

- *Challenge Phase:* The challenger, in this phase, flips a fair coin $a \leftarrow 0, 1$, and the adversary selects two queries Q_{I_a} and Q_{I_b}, which are then sent to the challenger. If the coin lands on heads, the challenger generates the trapdoor T_{Q_a} for Q_{I_a} and sends it back to the adversary.

- *Outcome Phase:* To win the challenge, the adversary must correctly guess which query, *i.e.,* either Q_{I_a} or Q_{I_b}, is associated with the received trapdoor, with a probability greater than 1/2. If the adversary cannot make such a guess with high probability, the scheme is considered secure with respect to Query-Trapdoor Indistinguishability.

Let KG, E_s, Enc, TrG, $SimSearch$, D_s be a similar image-based homomorphic SE scheme over a set of images I_i, facial image hashes H_{F_i}, query image Q_I, security parameter λ and adversary $\mathcal{A}$ over *'N'* number of facial images respectively. A probabilistic experimental function is as follows:

$$(s_k, p_k) \leftarrow KG(pb)$$

$$E_{I_i} \leftarrow E_s(I_i, s_k)$$

$$E_{H_{F_i}} \leftarrow Enc(p_k, H_{F_i})$$

$$for\ 0 < i < N:$$

$$(s_{\mathcal{A}}, Q_{I_i}) \leftarrow \mathcal{A}(s_{\mathcal{A}}, T_{Q_1}, T_{Q_2}, ...T_{Q_i})$$

$$T_{Q_i} \leftarrow TrG(Q_{I_i}, p_k)$$

$$a \leftarrow \{0,1\};$$

$$(s_{\mathcal{A}}, Q_{I_0}, Q_{I_1} \leftarrow \mathcal{A}(s_k, p_k)$$

$$T_{Q_a} \leftarrow TrG(Q_{I_i}, p_k)$$

$$a' \leftarrow \mathcal{A}_{N+1}(s_{\mathcal{A}}, T_{Q_a})$$

$$T_{Q'_a} \leftarrow TrG(Q_{I_j}, p_k); j \in N$$

$$if a' = a;\ output\ 1;$$

$$otherwise\ output\ 0$$

where $s_{\mathcal{A}}$ represents the state of the adversary $\mathcal{A}$. Such a scheme can be pronounced secure with respect to Query-Trapdoor Indistinguishability if the probability remains less than 1/2.

The proposed scheme builds upon the key generation and encryption phases of the Paillier cryptosystem. It generates distinct encrypted facial image hashes through probabilistic encryption and creates a unique trapdoor for each query, even if the

query is repeated. A probabilistic search algorithm produces the most similar image to the trapdoor via a predefined threshold value, making it impossible for the adversary $\mathcal{A}$ to accurately guess the original image or the facial image searched from an encrypted result. Additionally, both the adversary $\mathcal{A}$ and CS are unable to predict or deduce the search pattern. Thus, as a result of the probabilistic trapdoors, the proposed scheme satisfies the security definitions of adaptive & non-adaptive security, as well as Query-Trapdoor indistinguishability.

5.3.2 Game 2: Trapdoor-Image Indistinguishability

Suppose there exist multiple trapdoors, denoted as T_{Q_1}, T_{Q_2}, ..., T_{Q_i}, for all the images I_i stored at CS. The game involves the following three phases between an adversary and a challenger:

- *Query Phase:* The process begins with the challenger generating encrypted trapdoors against the image data I_i. Next, the adversary sends an encrypted trapdoor T_{Q_i} to the challenger, who responds by returning the most similar image according to a predefined similarity threshold. This cycle of queries and corresponding responses goes on until the adversary has collected a polynomial number of trapdoor-image pairs.

- *Challenge Phase:* The adversary selects two new trapdoors T_{Q_a} and T_{Q_b} and sends them over to the challenger. The challenger flips a fair coin $a \leftarrow 0, 1$, and performs a search among the encrypted images to select the most similar image E_{I_i}, which is then sent back to the adversary.

- *Outcome Phase:* The adversary $\mathcal{A}$ then has to make a guess about the resultant trapdoor a or b with a probability of higher than 50% otherwise the scheme is termed be attain Trapdoor-Image Indistinguishability.

Let $KG, E_s, Enc, TrG, SimSearch, D_s$ be a similar image-based homomorphic SE scheme over a set of images I_i, facial image hashes H_{F_i}, security parameter λ and adversary $\mathcal{A}$ over *'M'* number of images:

$$(s_k, p_k) \leftarrow KG(pb)$$

$$E_{I_i} \leftarrow E_s(I_i, s_k)$$

$$E_{H_{F_i}} \leftarrow Enc(p_k, H_{F_i})$$

$$for\ 0 < i < M:$$

$$(s_{\mathcal{A}}, T_{Q_i}) \leftarrow \mathcal{A}(s_{\mathcal{A}}, I_1, I_2, ...I_i)$$

$$I_i \leftarrow SimSearch(E_{H_{F_i}}, T_{Q_i})$$

$$a \leftarrow \{0, 1\};$$

$$(s_{\mathcal{A}}, T_{Q_0}, T_{Q_1} \leftarrow \mathcal{A}(I_i, p_k)$$

$$I_a \leftarrow SimSearch(E_{H_{F_a}}, T_{Q_a})$$

$$a' \leftarrow \mathcal{A}_{N+1}(s_{\mathcal{A}}, I_a)$$

$$T_{Q'_a} \leftarrow TrG(Q_{I_j}, p_k); j \in N$$

$$if a' = a;\ output\ 1;$$

$$otherwise\ output\ 0$$

where $s_{\mathcal{A}}$ represents the state of the adversary $\mathcal{A}$. The scheme can be pronounced secure with respect to Trapdoor-Image Indistinguishability if the probability remains less than 1/2.

The proposed scheme conducts searches as per the similarity of the encrypted images stored at CS, where trapdoors are generated probabilistically that ensure that even for the same query, multiple distinct trapdoors are generated in every iteration. Additionally, it is nearly impossible for the adversary $\mathcal{A}$ to establish a query-trapdoor-image connection, even if the adversary tracks the search history and outcomes. The scheme's use of probabilistic encryption to generate facial image hash values and trapdoors, with each encrypted trapdoor being unique, means that the likelihood of the adversary predicting the correct outcome is always less than 1/2. Therefore, the proposed scheme satisfies the security definitions of search pattern security and Trapdoor-Image Indistinguishability.

Since the scheme is proven to be secure with respect to both *Query - Trapdoor Indistinguishability* and *Trapdoor - Image Indistinguishability*, it implies that the scheme is IND-CPA secure as well.

5.3.3 Leakages

In a typical scenario, it is assumed that the adversary $\mathcal{A}$ launches the attack and is not constrained by using any weak structure in place of the proposed scheme. The focus of the leakages discussed below is the information that is revealed in polynomial time:

- **Leakage** L_1: It is linked to information kept on CS, specifically the number of encrypted image data and encrypted facial image hashes. All the images are outsourced to CS after encryption, so CS can only know just the quantity and not the underlying plaintexts.

$$L_1 = \left\{ E_{I_i}, E_{H_{F_i}}, (number\ of\ E_{I_i}), (number\ of\ E_{H_{F_i}}) \right\}$$

- **Leakage** L_2: It is linked to the generation of trapdoors from the queries. By using Paillier encryption, the trapdoor is probabilistically created and offers no insight into the underlying query being made at any point in the proposed scheme.

$$L_2 = \left\{ ((g^{Q_I}) * (r^n))(mod n^2) \right\}$$

- **Leakage** L_3: It is connected to the final outcome of the proposed searchable encryption scheme. The search is conducted at CS, and all authorized entities and the adversary $\mathcal{A}$, can have access to the results. The search results are encrypted after the result of an "addition" function, and only the data owner/ authorized user (possessing the secret key) can decrypt them. They do not contain any details about the underlying search terms or queries.

$$L_3 = \left\{ A_{DD}(T_I, E_{H_{F_i}}), (V_R) \right\}$$

The assumptions and leakages mentioned above are linked and rely on one another. As a result, in order to achieve the maximum degree of security, it is necessary to strictly adhere to all security assumptions. Additionally, none of the leaks are revealing the plaintext or any details on the properties of the plaintext; as a result, the suggested method is strong and adheres to security requirements. According to the corollary presented in [116], such a scheme can be termed a privacy-preserving searchable encryption scheme.

5.3.4 Soundness

Soundness refers to the property that an adversary cannot produce any false positive search results, *i.e.*, the scheme does not return any result that does not match the search query. In other words, if the scheme returns a search result, it is guaranteed to be a valid match for the search query. A searchable scheme that lacks soundness may be susceptible to security threats aiding an adversary to get access to critical information by submitting a carefully crafted search query that returns false positives.

For the proposed scheme to be deemed sound, it must ensure that the security parameters (g, λ, μ) and the key pair (p_k, s_k) used to encrypt facial hash values $E_{H_{F_i}}$ via encryption function $Enc(p_k, H_{F_i})$, as well as the search process using trapdoors (T_Q), do not generate false positives and always produce meaningful search results with a high degree of probability.

5.3.5 Correctness

Correctness, on the other hand, refers to the property that the scheme returns all the valid search results, *i.e.*, it does not miss any match for the search query. A searchable scheme that lacks correctness may fail to return valid search results leading to data loss or privacy threats.

The correctness of the proposed approach can be verified by ensuring that the security parameters (g, λ, μ) and the key pair (p_k, s_k) used to encrypt facial hashes $(E_{H_{F_i}})$ through the function $Enc(p_k, H_{F_i})$, as well as the search process using trapdoors T_Q, consistently yield similar images with a high degree of probability.

Thus, the proposed homomorphic-based similar image SE scheme possesses the properties of both soundness and correctness and thus, ensures the confidentiality and integrity of the encrypted data.

5.4 Security Attributes Comparison

The section presents a comprehensive comparison among many schemes with respect to different security attributes. Homomorphic encryption entails that the processing can be carried out on encrypted data with similar results as if done on the underlying plaintexts without revealing any information about the plaintexts. Index-based implies that the scheme is reliant on ranking/indices or lookup tables that can lead the CS to gain information about the trapdoor-image relation. The lack of any index and probabilistic trapdoors will ensure search pattern security.

No leakage of information about the data will lead to a scheme providing privacy preservation. Table 5.1 presents the security attributes comparison.

Table 5.1: Comparison of Security Attributes

Attributes	HE based	Index based	Probabilistic Trapdoors	Secure Search Pattern	Preserving Privacy
[82]		✓			
[84]	✓	✓			✓
[85]	✓		✓	✓	✓
[88]	✓				✓
[92]		✓			
[93]		✓			
[94]		✓			✓
[97]		✓			
[86]	✓				✓
[99]		✓			
[101]	✓	✓			✓
[102]		✓			✓
[103]	✓			✓	✓
[107]					✓
[108]		✓	✓	✓	✓
[109]		✓			✓
[110]		✓			✓
[111]	✓				✓
[105]		✓		✓	✓
[104]		✓		✓	✓
[112]	✓		✓	✓	✓
[115]	✓				✓
[113]	✓		✓		✓
PS	✓		✓	✓	✓

It is clear from the table 5.1 that while most of the schemes provide privacy preservation, many are susceptible to secure search pattern attacks. Schemes presented in [85, 108, 112, 113] generate probabilistic trapdoors. The scheme [112] is based on HE but the proposed scheme follows a pixel-by-pixel match of images. The proposed privacy-preserving scheme presented in this research is based on homomorphic encryption, generates probabilistic trapdoors thus ensuring secure search patterns, and carries out searching at run-time without the maintenance of

any index table.

5.5 Summary

In this chapter, the security definitions were revisited in detail. The security analysis was presented in terms of the aforementioned definitions, leakages, soundness, and correctness of the proposed scheme. Lastly, a comparison among the latest schemes was drawn with respect to different security attributes. The performance analysis is carried out in chapter 6.

Chapter 6

Performance Analysis

6.1 Overview

The chapter presents a comprehensive performance analysis of the proposed work in three major parts. Firstly, the performance metrics are defined and explained in terms of the computing time required to run a protocol and are displayed graphically. Secondly, the storage overhead of the proposed scheme is presented. Finally, the computational complexity of our proposed scheme is described in comparison with the latest schemes.

6.2 Dataset Description

The dataset used for the testing and the performance analysis of our proposed work is *Chokepoint Dataset* [117]. The dataset is developed for studies in person identification in a real-world surveillance scenario situation. An array of three

Figure 6.1: Multiple Images from Chokepoint Dataset

cameras was set above multiple doors to film persons going naturally through them. A lot of facial pictures were taken when a person was entering through a door or an enclosed space. The images were saved in *jpg* format. It was implied in the making of the dataset that faces in such cases will differ in terms of sharpness, lighting, and misalignment owing to automated face position depending on a person's posture. The dataset is optimal to use for a similar image search scheme as is done in our proposed work. Figure 6.1 show some of the images from the dataset. It shows that many images are showing no person and thus no detectable facial

image, while some have more than one person in line with the camera lens. Some faces have been captured properly, whereas few have been captured in a different posture failing to show clear-cut facial features.

6.3 Performance Metrics

The simulations for our proposed work were carried out in a client-server model where the AES encryption of the images, Paillier HE of the facial hash values, generation of trapdoor, and finally, the decryption of retrieved image(s) are carried out by the data owner at client side and searching is carried out the cloud server. Both the client and server-side simulations were carried out on a system of core *i7*, *7th* generation with 16 GB RAM and 256 GB SSD running Ubuntu OS 18.04.5 LTS (64 bits). The testing was carried out over 500 images as well as up to 500 facial images from the chokepoint dataset. The facial images were retrieved from the dataset via MTCNN [60]. After that, the perceptual hash values were calculated for all facial images.

The AES encryption of images was done in iterations of 100 images. The time taken for AES encryption of 100, 200, 300, 400, and 500 images came out to be 3.91 sec, 9.2 sec, 12.8 sec, 17.94 sec, and 21.19 sec respectively. The images in *jpg* format were saved in *png* format after AES encryption so as to retain the images' original features. The results of image encryption via AES are shown graphically in figure 6.2 where the x-axis shows number of images and time (in seconds) is plotted at y-axis.

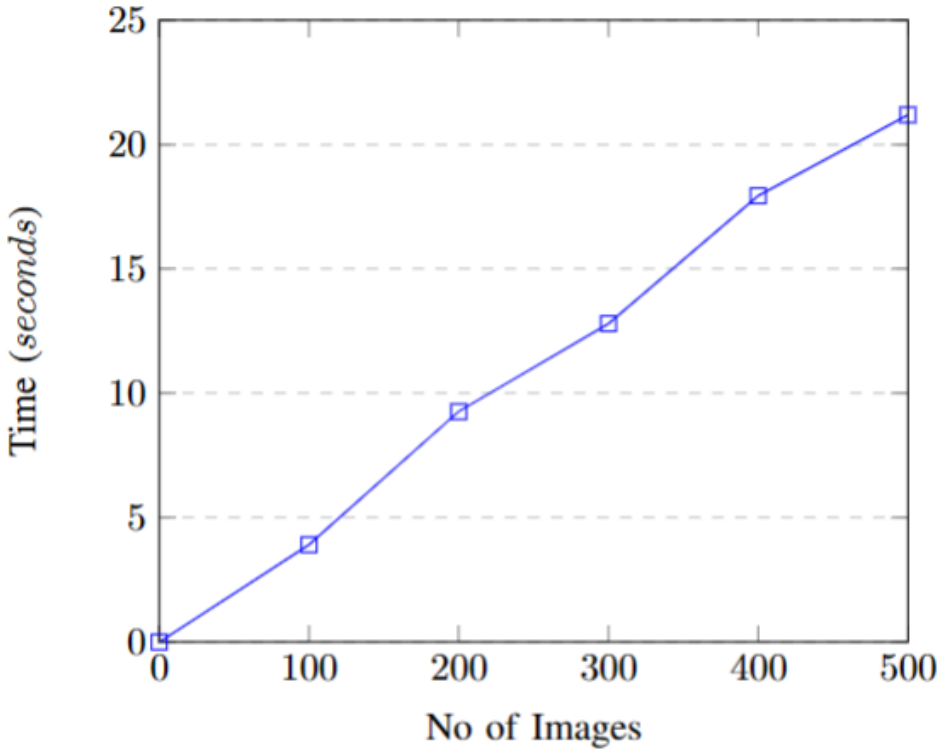


Figure 6.2: Images Encryption Time (Standard Encryption-AES)

The next part of the scheme included the encryption of facial image hash values by Paillier HE. It was carried out for 500 facial image hash values, first with iterations of 10 up to 50 and then 100 up to 500 images. The time taken for image hash calculation and Paillier HE is presented in the table 6.1 and shown graphically in figure 6.3. The graph includes number of image hash values and time taken (in seconds) plotted on x-axis and y-axis respectively.

The phase of a similar search was carried out via trapdoor generation and using properties of scalar multiplication and addition. The time taken to generate the trapdoor was 14 milliseconds and searching was carried out over 500 facial image hashes in iterations of 10 up to 50 and then 100 up to 500 hashes. The time taken for similar image search is presented in table 6.2 and the results are shown

Table 6.1: Facial Image Hash Encryption (Paillier HE) Time

No. of Facial Image Hashes	PHE Encryption Time (sec)
10	0.057
20	0.141
30	0.252
40	0.296
50	0.428
100	0.659
200	1.988
300	2.301
400	3.291
500	4.112

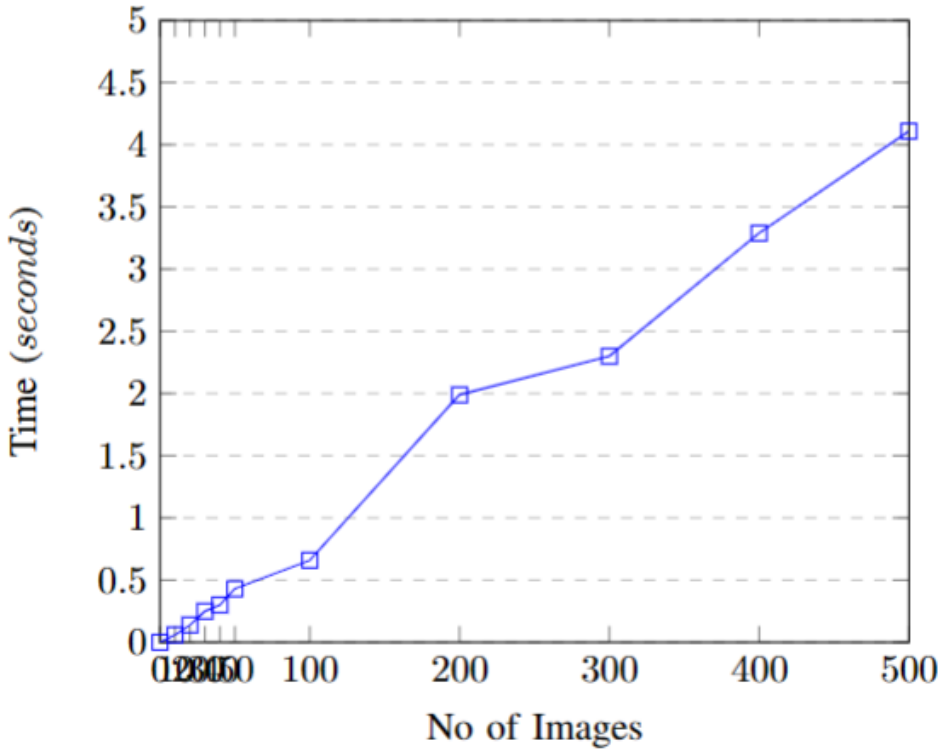


Figure 6.3: Facial Image Hash Encryption (Paillier HE) Time

graphically in figure 6.4. The graph includes number of image hash values and time taken (in seconds) plotted on x-axis and y-axis respectively.

The AES decryption of images was done in iterations of 100 images, similar to AES encryption phase. The time taken for AES decryption of 100, 200, 300, 400,

Table 6.2: Similar Image Searching Time

No. of Facial Image Hashes	Time (seconds)
10	0.018
20	0.037
30	0.0.056
40	0.0.069
50	0.095
100	0.171
200	0.361
300	0.582
400	0.783
500	0.988

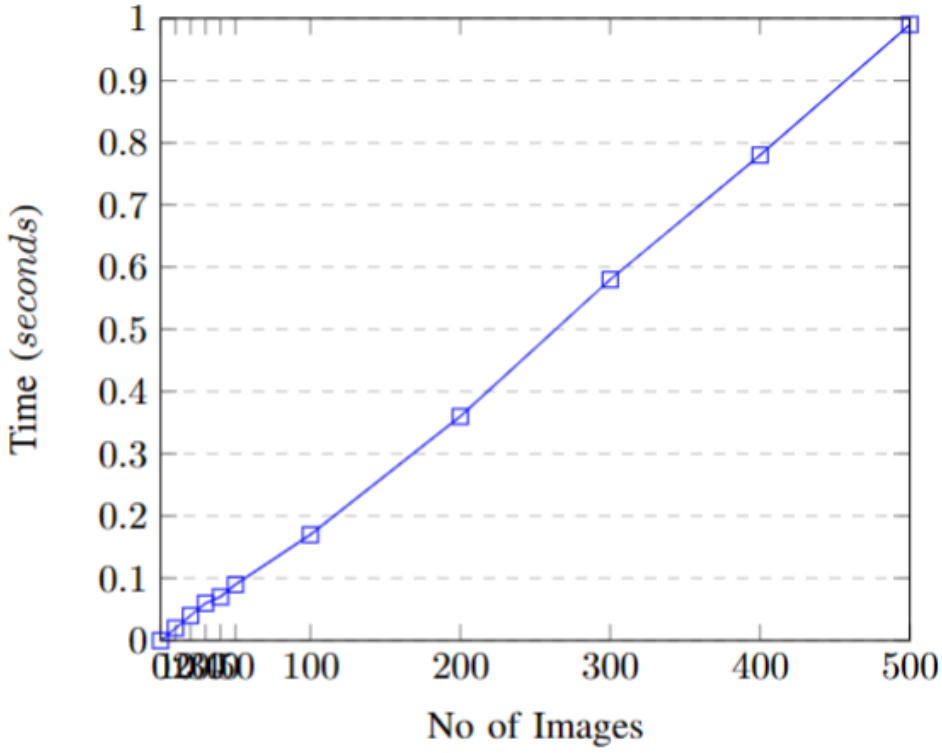


Figure 6.4: Similar Image Searching Time

and 500 images came out to be 1.78 sec, 3.67 sec, 6.73 sec, 7.91 sec, and 11.04 sec respectively. The results of image encryption via AES are shown graphically in figure 6.5 where the x-axis shows number of images and time (in seconds) is plotted at y-axis.

Lastly, a comparison of execution time and storage overhead was drawn for our

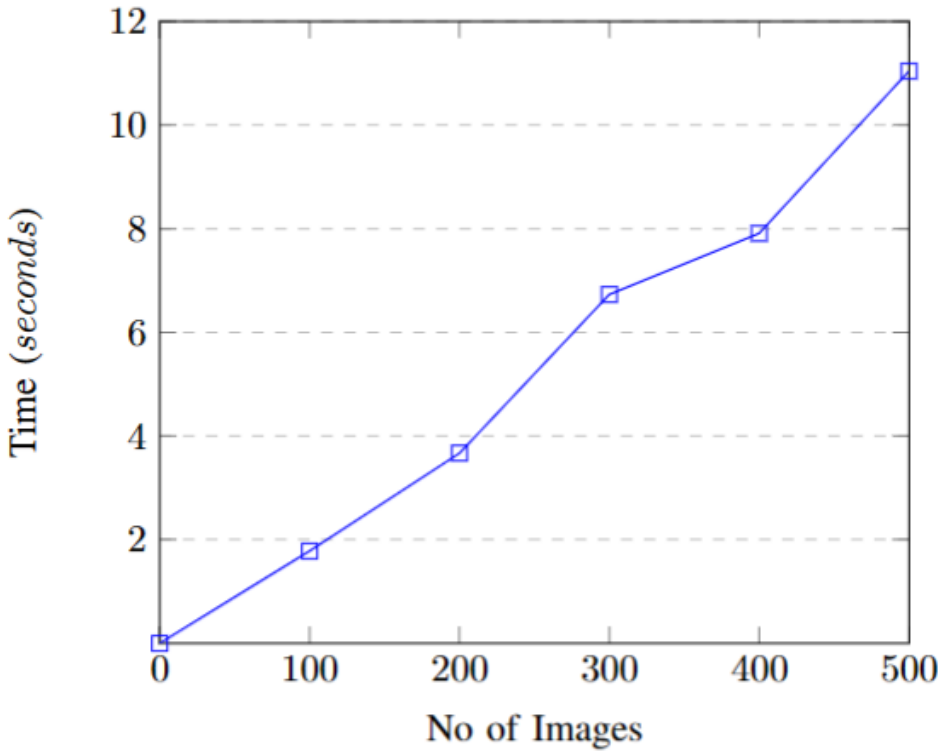


Figure 6.5: Images Decryption Time (Standard Decryption-AES)

scheme with another image-based homomorphic SE scheme [112]. The authors in [112] ran simulations for their scheme on a single image with a single object. They claimed that their scheme ran for 42.766 seconds and took up storage space of 2.54 MB in total. In comparison, a single image with a single detectable face of size 47.9 kB was encrypted with AES, and a size increase of 1.4 MB was observed. The AES encryption was executed in 0.046 seconds. After that, the perceptual hash was calculated for the facial image and it was encrypted via Paillier HE taking 0.01 seconds. For searching, the trapdoor generation and searching took 14 and 5 milliseconds respectively. The image retrieved was decrypted in 0.015 seconds. The proposed scheme took a total of 65 milliseconds with a storage overhead of 1.4 MB overall.

6.4 Storage Overhead

The storage overhead indicates the increase in the size of images after AES encryption. The values of sizes before and after AES encryption are shown in the table 6.3 for up to 500 images. It is evident from the table entries that there is a drastic change in the sizes of the images such that for 100 images, the cumulative size is 5.4 MB which translates to 137.6 Mb after AES encryption making it 2448% increase. Similarly, for 500 images, the increase in size is approximately 2578%. This shows the needful for 3rd party storage services *i.e.,* cloud server as local management and storage of such huge data is very difficult and poses many security threats as well.

Table 6.3: Storage Overhead

No. of Images	Before AES Encryption	After AES Encryption
100	5.4 MB	137.6 MB
200	10.6 MB	275.3 MB
300	15.9 MB	412.9 MB
400	21.3 MB	550.5 MB
500	25.7 MB	688.2 MB

6.5 Computational Complexity

The section presents a comparison of the computational complexity of our proposed scheme for some of the latest SE schemes. Table 6.4 shows the computational complexity of our proposed scheme against state-of-the-art SE schemes. For all algorithms, the complexity is defined in terms of asymptotic notations such that M shows the total number of images, N and F represent the image objects

and facial hashes respectively while P denotes the number of pixels in an image and finally, S is used for the classes of different image data. $\mathbf{O}(1)$ represents the complexity for hash functions used for trapdoor generation functions employed by different schemes of [97], [112] as well as our proposed work.

Table 6.4: Computational Complexity

Phases	[97]	[101]	[106]	[112]	**PS**
Key Generation	$\mathcal{O}(2^{\lambda+1})$	$\mathcal{O}(2^{\lambda})$	$\mathcal{O}(2^{\lambda+1})$	$\mathcal{O}(2^{\lambda})$	$\mathcal{O}(2^{\lambda})$
Feature Extraction Object Detection	$\mathcal{O}(N)$	$\mathcal{O}(M.N)$	-	$\mathcal{O}(N.P)$	$\mathcal{O}(N)$
Image Encryption	$\mathcal{O}(M)$	$\mathcal{O}(M)$	$\mathcal{O}(M^2+1)$	$\mathcal{O}(M)$	$\mathcal{O}(M)$
Index Generation	$\mathcal{O}(N)$	$\mathcal{O}(M.N)$	$\mathcal{O}(8M.S^2)$	-	-
Trapdoor Generation	$\mathcal{O}(1)$	$\mathcal{O}(N)$	$\mathcal{O}(4S^2)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$
Searching	$\mathcal{O}(2N)$	$\mathcal{O}(2.M.N)$	$\mathcal{O}(4S^2+2S.M)$	$\mathcal{O}(M.N.P)$	$\mathcal{O}(M.N)$
Image Decryption	$\mathcal{O}(M)$	$\mathcal{O}(M)$	$\mathcal{O}(M^2+1)$	$\mathcal{O}(M)$	$\mathcal{O}(M)$

6.6 Summary

The chapter presented a comprehensive performance analysis of the proposed work. The performance analysis was discussed in three major sections. Firstly, the performance metrics were defined and explained in terms of the computing time required to run a protocol and were displayed graphically. Secondly, the storage overhead of the proposed scheme was presented. Finally, the computational complexity of our proposed scheme was presented in comparison with some of the latest schemes. The research is finally concluded in chapter 7.

Chapter 7

Conclusion

Surveillance data images of crowded places have become increasingly important in today's world for a variety of reasons. Cloud-based image searching schemes leverage distributed computing power to search large amounts of data quickly and accurately, allowing users to easily find relevant images based on their search queries. A novel scheme for similar image searching based on homomorphic encryption is proposed for cloud-connected devices. The scheme is based on probabilistic trapdoors and ensures the privacy preservation of image data. Implementation and testing of the proposed scheme are carried out over a real-world data set in order to assess its security and performance in terms of complexity, computation, and storage overheads. The efficiency and better testing of the scheme can be done by its deployment over a real Cloud platform as well as by introducing parallel processing. Although the research is novel in its context, the factor of human dependency can be eradicated by the development of a fog/edge layer in the scheme and can be regarded as potential future works.

References

[1] Lynsey Dubbeld. Observing bodies. camera surveillance and the significance of the body. Ethics and Information technology, 5(3):151, 2003.

[2] Zheng Xu, Chuanping Hu, and Lin Mei. Video structured description technology based intelligence analysis of surveillance videos for public security applications. Multimedia Tools and Applications, 75:12155–12172, 2016.

[3] Wei Qi Yan. Introduction to intelligent surveillance: surveillance data capture, transmission, and analytics. Springer, 2019.

[4] Yassir Zardoua, Abdelali Astito, and Mohammed Boulaala. A comparison of ais, x-band marine radar systems and camera surveillance systems in the collection of tracking data. arXiv preprint arXiv:2206.12809, 2022.

[5] G Sreenu and Saleem Durai. Intelligent video surveillance: a review through deep learning techniques for crowd analysis. Journal of Big Data, 6(1):1–27, 2019.

[6] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype,

and reality for delivering computing as the 5th utility. Future Generation computer systems, 25(6):599–616, 2009.

[7] Ado Adamou Abba Ari, Olga Kengni Ngangmo, Chafiq Titouna, Ousmane Thiare, Alidou Mohamadou, and Abdelhak Mourad Gueroui. Enabling privacy and security in cloud of things: Architecture, applications, security & privacy challenges. Applied Computing and Informatics, 2020.

[8] Ling Du, Wei Zhang, Huazhu Fu, Wenqi Ren, and Xinpeng Zhang. An efficient privacy protection scheme for data security in video surveillance. Journal of Visual Communication and Image Representation, 59:347–362, 2019.

[9] Kui Ren, Cong Wang, and Qian Wang. Security challenges for the public cloud. IEEE Internet computing, 16(1):69–73, 2012.

[10] Zhangjie Fu, Kui Ren, Jiangang Shu, Xingming Sun, and Fengxiao Huang. Enabling personalized search over encrypted outsourced data with efficiency improvement. IEEE transactions on parallel and distributed systems, 27(9): 2546–2559, 2015.

[11] Kaitai Liang and Willy Susilo. Searchable attribute-based mechanism with efficient data sharing for secure cloud storage. IEEE Transactions on Information Forensics and Security, 10(9):1981–1992, 2015.

[12] Pan Jun Sun. Privacy protection and data security in cloud computing: a survey, challenges, and solutions. IEEE Access, 7:147420–147452, 2019.

[13] David Archer, Lily Chen, Jung Hee Cheon, Ran Gilad-Bachrach, Roger A Hallman, Zhicong Huang, Xiaoqian Jiang, Ranjit Kumaresan, Bradley A Malin, Heidi Sofia, et al. Applications of homomorphic encryption. HomomorphicEncryption. org, Redmond WA, Tech. Rep., 2017.

[14] Fei Han, Jing Qin, and Jiankun Hu. Secure searches in the cloud: A survey. Future Generation Computer Systems, 62:66–75, 2016.

[15] Hoang Pham, Jason Woodworth, and Mohsen Amini Salehi. Survey on secure search over encrypted data on the cloud. Concurrency and Computation: Practice and Experience, 31(17):e5284, 2019.

[16] Khadijah Chamili, Md Jan Nordin, Waidah Ismail, and Abduljalil Radman. Searchable encryption: A review. International Journal of Security and Its Applications, 11:79–88, 2017.

[17] Dawn Xiaoding Song, David Wagner, and Adrian Perrig. Practical techniques for searches on encrypted data. In Proceeding 2000 IEEE Symposium on Security and Privacy. S&P 2000, pages 44–55. IEEE, 2000.

[18] Dan Boneh, Giovanni Di Crescenzo, Rafail Ostrovsky, and Giuseppe Persiano. Public key encryption with keyword search. In International conference on the theory and applications of cryptographic techniques, pages 506–522. Springer, 2004.

[19] Adi Shamir. Identity-based cryptosystems and signature schemes. In Workshop on the theory and application of cryptographic techniques, pages 47–53. Springer, 1984.

[20] Xu An Wang, Fatos Xhafa, Weiyi Cai, Jianfeng Ma, and Fushan Wei. Efficient privacy-preserving predicate encryption with fine-grained searchable capability for cloud storage. Computers & Electrical Engineering, 56:871–883, 2016.

[21] Jonathan Katz, Amit Sahai, and Brent Waters. Predicate encryption supporting disjunctions, polynomial equations, and inner products. In annual international conference on the theory and applications of cryptographic techniques, pages 146–162. Springer, 2008.

[22] Ning Cao, Cong Wang, Ming Li, Kui Ren, and Wenjing Lou. Privacy-preserving multi-keyword ranked search over encrypted cloud data. IEEE Transactions on parallel and distributed systems, 25(1):222–233, 2013.

[23] Rafail Ostrovsky and William E Skeith. A survey of single-database private information retrieval: Techniques and applications. In International Workshop on Public Key Cryptography, pages 393–411. Springer, 2007.

[24] Caroline Fontaine and Fabien Galand. A survey of homomorphic encryption for nonspecialists. EURASIP Journal on Information Security, 2007:1–10, 2007.

[25] Xun Yi, Russell Paulet, and Elisa Bertino. Homomorphic encryption. In Homomorphic Encryption and Applications, pages 27–46. Springer, 2014.

[26] Maha Tebaa, Saïd El Hajji, and Abdellatif El Ghazi. Homomorphic encryption applied to the cloud computing security. In Proceedings of the World Congress on Engineering, volume 1, pages 4–6, 2012.

[27] Payal V Parmar, Shraddha B Padhar, Shafika N Patel, Niyatee I Bhatt, and Rutvij H Jhaveri. Survey of various homomorphic encryption algorithms and schemes. International Journal of Computer Applications, 91(8), 2014.

[28] Lifang Zhang, Yan Zheng, and Raimo Kantoa. A review of homomorphic encryption and its applications. In Proceedings of the 9th EAI International Conference on Mobile Multimedia Communications, pages 97–106, 2016.

[29] Caroline Fontaine and Fabien Galand. A survey of homomorphic encryption for nonspecialists. EURASIP Journal on Information Security, 2007:1–10, 2007.

[30] Joan Daemen and Vincent Rijmen. Aes proposal: Rijndael. 1999.

[31] Abdullah Al Hasib and Abul Ahsan Md Mahmudul Haque. A comparative study of the performance and security issues of aes and rsa cryptography. In 2008 Third International Conference on Convergence and Hybrid Information Technology, volume 2, pages 505–510. IEEE, 2008.

[32] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In International conference on the theory and applications of cryptographic techniques, pages 223–238. Springer, 1999.

[33] Ramarathnam Venkatesan, S-M Koon, Mariusz H Jakubowski, and Pierre Moulin. Robust image hashing. In Proceedings 2000 International Conference on Image Processing (Cat. No. 00CH37101), volume 3, pages 664–666. IEEE, 2000.

[34] Ashwin Swaminathan, Yinian Mao, and Min Wu. Robust and secure image hashing. IEEE Transactions on Information Forensics and security, 1(2): 215–230, 2006.

[35] Jingdong Wang, Heng Tao Shen, Jingkuan Song, and Jianqiu Ji. Hashing for similarity search: A survey. arXiv preprint arXiv:1408.2927, 2014.

[36] Rafael Padilla, Sergio L Netto, and Eduardo AB Da Silva. A survey on performance metrics for object-detection algorithms. In 2020 international conference on systems, signals and image processing (IWSSIP), pages 237–242. IEEE, 2020.

[37] Zhengxia Zou, Keyan Chen, Zhenwei Shi, Yuhong Guo, and Jieping Ye. Object detection in 20 years: A survey. Proceedings of the IEEE, 2023.

[38] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014.

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9):1904–1916, 2015.

[40] Ross Girshick. Fast r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 1440–1448, 2015.

[41] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2117–2125, 2017.

[42] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. Advances in neural information processing systems, 28:91–99, 2015.

[43] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In Proceedings of the IEEE international conference on computer vision, pages 2961–2969, 2017.

[44] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 779–788, 2016.

[45] Joseph Redmon and Ali Farhadi. Yolo9000: better, faster, stronger. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 7263–7271, 2017.

[46] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. Ssd: Single shot multibox detector. In European conference on computer vision, pages 21–37. Springer, 2016.

[47] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Fo-

cal loss for dense object detection. In Proceedings of the IEEE international conference on computer vision, pages 2980–2988, 2017.

[48] Jifeng Dai, Yi Li, Kaiming He, and Jian Sun. R-fcn: Object detection via region-based fully convolutional networks. In Advances in neural information processing systems, pages 379–387, 2016.

[49] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767, 2018.

[50] Junxiao Li and Ziao Wu. The application of yolov4 and a new pedestrian clustering algorithm to implement social distance monitoring during the covid-19 pandemic. In Journal of Physics: Conference Series, volume 1865, page 042019. IOP Publishing, 2021.

[51] Piotr Dollár, Ron Appel, Serge Belongie, and Pietro Perona. Fast feature pyramids for object detection. IEEE transactions on pattern analysis and machine intelligence, 36(8):1532–1545, 2014.

[52] Alex Bewley, Zongyuan Ge, Lionel Ott, Fabio Ramos, and Ben Upcroft. Simple online and realtime tracking. In 2016 IEEE international conference on image processing (ICIP), pages 3464–3468. IEEE, 2016.

[53] Thomas Fortmann, Yaakov Bar-Shalom, and Molly Scheffe. Sonar tracking of multiple targets using joint probabilistic data association. IEEE journal of Oceanic Engineering, 8(3):173–184, 1983.

[54] Alexey Bochkovskiy, Chien-Yao Wang, and Hong-Yuan Mark Liao.

Yolov4: Optimal speed and accuracy of object detection. arXiv preprint arXiv:2004.10934, 2020.

[55] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In Proceedings of the 2001 IEEE computer society conference on computer vision and pattern recognition. CVPR 2001, volume 1, pages I–I. Ieee, 2001.

[56] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. 1:886–893, 2005.

[57] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. IEEE transactions on pattern analysis and machine intelligence, 32(9):1627–1645, 2009.

[58] Pedro F Felzenszwalb. Discriminatively trained deformable part models, release 4. http://people. cs. uchicago. edu/pff/latent-release4/, 2010.

[59] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 815–823, 2015.

[60] Kaipeng Zhang, Zhanpeng Zhang, Zhifeng Li, and Yu Qiao. Joint face detection and alignment using multitask cascaded convolutional networks. IEEE signal processing letters, 23(10):1499–1503, 2016.

[61] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 6154–6162, 2018.

[62] Mingxing Tan, Ruoming Pang, and Quoc V Le. Efficientdet: Scalable and efficient object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 10781–10790, 2020.

[63] Kaiwen Duan, Song Bai, Lingxi Xie, Honggang Qi, Qingming Huang, and Qi Tian. Centernet: Keypoint triplets for object detection. In Proceedings of the IEEE/CVF international conference on computer vision, pages 6569–6578, 2019.

[64] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. Freeanchor: Learning to match anchors for visual object detection. Advances in neural information processing systems, 32, 2019.

[65] Chenchen Zhu, Yihui He, and Marios Savvides. Feature selective anchor-free module for single-shot object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 840–849, 2019.

[66] Rohit Mohan and Abhinav Valada. Efficientps: Efficient panoptic segmentation. International Journal of Computer Vision, 129(5):1551–1579, 2021.

[67] Xianzhi Du, Tsung-Yi Lin, Pengchong Jin, Golnaz Ghiasi, Mingxing Tan, Yin Cui, Quoc V Le, and Xiaodan Song. Spinenet: Learning scale-permuted backbone for recognition and localization. In Proceedings of the IEEE/CVF

conference on computer vision and pattern recognition, pages 11592–11601, 2020.

[68] Ze Yang, Shaohui Liu, Han Hu, Liwei Wang, and Stephen Lin. Reppoints: Point set representation for object detection. In Proceedings of the IEEE/CVF international conference on computer vision, pages 9657–9666, 2019.

[69] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. pages 213–229, 2020.

[70] Peize Sun, Rufeng Zhang, Yi Jiang, Tao Kong, Chenfeng Xu, Wei Zhan, Masayoshi Tomizuka, Lei Li, Zehuan Yuan, Changhu Wang, et al. Sparse r-cnn: End-to-end object detection with learnable proposals. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 14454–14463, 2021.

[71] Xizhou Zhu, Weijie Su, Lewei Lu, Bin Li, Xiaogang Wang, and Jifeng Dai. Deformable detr: Deformable transformers for end-to-end object detection. arXiv preprint arXiv:2010.04159, 2020.

[72] Upesh Nepal and Hossein Eslamiat. Comparing yolov3, yolov4 and yolov5 for autonomous landing spot detection in faulty uavs. Sensors, 22(2):464, 2022.

[73] Alexander Kirillov, Ross Girshick, Kaiming He, and Piotr Dollár. Panoptic

feature pyramid networks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 6399–6408, 2019.

[74] Chuyi Li, Lulu Li, Hongliang Jiang, Kaiheng Weng, Yifei Geng, Liang Li, Zaidan Ke, Qingyuan Li, Meng Cheng, Weiqiang Nie, et al. Yolov6: A single-stage object detection framework for industrial applications. arXiv preprint arXiv:2209.02976, 2022.

[75] Chien-Yao Wang, Alexey Bochkovskiy, and Hong-Yuan Mark Liao. Yolov7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv preprint arXiv:2207.02696, 2022.

[76] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 580–587, 2014.

[77] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Spatial pyramid pooling in deep convolutional networks for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 37(9):1904–1916, 2015.

[78] Yunling Wang, Jianfeng Wang, and Xiaofeng Chen. Secure searchable encryption: a survey. Journal of communications and information networks, 1 (4):52–65, 2016.

[79] Khadijah Chamili, Md Jan Nordin, Waidah Ismail, and Abduljalil Radman.

Searchable encryption: A review. International Journal of Security and Its Applications, 11:79–88, 2017.

[80] R Kirubakaramoorthi, D Arivazhagan, and D Helen. Survey on encryption techniques used to secure cloud storage system. Indian J. Sci. Technol, 8 (36):1–7, 2015.

[81] Hoang Pham, Jason Woodworth, and Mohsen Amini Salehi. Survey on secure search over encrypted data on the cloud. Concurrency and Computation: Practice and Experience, 31(17):e5284, 2019.

[82] Shengshan Hu, Qian Wang, Jingjun Wang, Zhan Qin, and Kui Ren. Securing sift: Privacy-preserving outsourcing computation of feature extractions over encrypted image data. IEEE Transactions on Image Processing, 25(7):3411–3425, 2016.

[83] Tony Lindeberg. Scale invariant feature transform. 2012.

[84] Chao-Yung Hsu, Chun-Shien Lu, and Soo-Chang Pei. Image feature extraction in encrypted domain with privacy-preserving sift. IEEE transactions on image processing, 21(11):4593–4607, 2012.

[85] Chao-Yung Hsu, Chun-Shien Lu, and Soo-Chang Pei. Image feature extraction in encrypted domain with privacy-preserving sift. IEEE transactions on image processing, 21(11):4593–4607, 2012.

[86] Zhan Qin, Jian Weng, Yong Cui, and Kui Ren. Privacy-preserving image processing in the cloud. IEEE cloud computing, 5(2):48–57, 2018.

[87] Rishav Chakravarti and Xiannong Meng. A study of color histogram based image retrieval. In 2009 Sixth International Conference on Information Technology: New Generations, pages 1323–1328. IEEE, 2009.

[88] Tengfei Yang, Jianfeng Ma, Qian Wang, Yinbin Miao, Xuan Wang, and Qian Meng. Image feature extraction in encrypted domain with privacy-preserving hahn moments. IEEE Access, 6:47521–47534, 2018.

[89] T Prabakar Joshua, M Arrivukannamma, and JGR Sathiaseelan. Comparison of dct and dwt image compression. International Journal of Emerging Trends in Science and Technology, Int. J of Computer Science and Mobile Computing, 5(4):62–67, 2016.

[90] Jian Zhou, Huazhong Shu, Hongqing Zhu, Christine Toumoulin, and Limin Luo. Image analysis by discrete orthogonal hahn moments. In International Conference Image Analysis and Recognition, pages 524–531. Springer, 2005.

[91] Hongqing Zhu, Huazhong Shu, Jian Zhou, Limin Luo, and Jean-Louis Coatrieux. Image analysis by discrete orthogonal dual hahn moments. Pattern Recognition Letters, 28(13):1688–1704, 2007.

[92] Yi Zhu, Xingming Sun, Zhihua Xia, and Naixue Xiong. Secure similarity search over encrypted cloud images. International Journal of Security and Its Applications, 9(8):1–14, 2015.

[93] Sayyada Fahmeeda Sultana and DC Shubhangi. Privacy preserving lbp based feature extraction on encrypted images. In 2017 International

Conference on Computer Communication and Informatics (ICCCI), pages 1–4. IEEE, 2017.

[94] Yuan Wang, Meixia Miao, Jian Shen, and Jianfeng Wang. Towards efficient privacy-preserving encrypted image search in cloud computing. Soft Computing, 23(6):2101–2112, 2019.

[95] Abelino Jiménez, Bhiksha Raj, Jose Portelo, and Isabel Trancoso. Secure modular hashing. In 2015 IEEE International Workshop on Information Forensics and Security (WIFS), pages 1–6, 2015. doi: 10.1109/WIFS.2015.7368567.

[96] Greg Hamerly and Charles Elkan. Learning the k in k-means. Advances in neural information processing systems, 16:281–288, 2004.

[97] Zhihua Xia, Neal N Xiong, Athanasios V Vasilakos, and Xingming Sun. Epcbir: An efficient and privacy-preserving content-based image retrieval scheme in cloud computing. Information Sciences, 387:195–204, 2017.

[98] Jianping He, Bin Liu, Deguang Kong, Xuan Bao, Na Wang, Hongxia Jin, and George Kesidis. Puppies: Transformation-supported personalized privacy preserving partial image sharing. In 2016 46th annual IEEE/IFIP international conference on dependable systems and networks (DSN), pages 359–370. IEEE, 2016.

[99] ZH Xia, LH Lu, Tong Qin, HJ Shim, XY Chen, and Byeungwoo Jeon. A privacy-preserving image retrieval based on ac-coefficients and color

histograms in cloud environment. CMC-COMPUTERS MATERIALS & CONTINUA, 58(1):27–43, 2019.

[100] Jiawei Yuan, Shucheng Yu, and Linke Guo. Seisa: Secure and efficient encrypted image search with access control. In 2015 IEEE conference on computer communications (INFOCOM), pages 2083–2091. IEEE, 2015.

[101] Lan Zhang, Taeho Jung, Kebin Liu, Xiang-Yang Li, Xuan Ding, Jiaxi Gu, and Yunhao Liu. Pic: Enable large-scale privacy preserving content-based image search on cloud. IEEE Transactions on Parallel and Distributed Systems, 28(11):3258–3271, 2017.

[102] Qin Zou, Jianfeng Wang, Jun Ye, Jian Shen, and Xiaofeng Chen. Efficient and secure encrypted image search in mobile cloud computing. Soft Computing, 21(11):2959–2969, 2017.

[103] Cheng Guo, Jing Jia, Kim-Kwang Raymond Choo, and Yingmo Jie. Privacy-preserving image search (ppis): Secure classification and searching using convolutional neural network over large-scale encrypted medical images. Computers & Security, 99:102021, 2020.

[104] Yingying Li, Jianfeng Ma, Yinbin Miao, Yue Wang, Tengfei Yang, Ximeng Liu, and Kim-Kwang Raymond Choo. Traceable and controllable encrypted cloud image search in multi-user settings. IEEE Transactions on Cloud Computing, 2020.

[105] Yating Duan, Yanping Li, Laifeng Lu, and Yong Ding. A faster out-

sourced medical image retrieval scheme with privacy preservation. Journal of Systems Architecture, 122:102356, 2022.

[106] Xiangyu Wang, Jianfeng Ma, Ximeng Liu, and Yinbin Miao. Search in my way: Practical outsourced image retrieval framework supporting unshared key. In IEEE INFOCOM 2019 - IEEE Conference on Computer Communications, pages 2485–2493, 2019. doi: 10.1109/INFOCOM.2019.8737619.

[107] R Punithavathi, A Ramalingam, Chinnarao Kurangi, A Reddy, and J Uthayakumar. Secure content based image retrieval system using deep learning with multi share creation scheme in cloud environment. Multimedia Tools and Applications, 80(17):26889–26910, 2021.

[108] Yating Duan, Yanping Li, Laifeng Lu, and Yong Ding. A faster outsourced medical image retrieval scheme with privacy preservation. Journal of Systems Architecture, 122:102356, 2022.

[109] Anyu Du, Liejun Wang, Shuli Cheng, and Naixiang Ao. A privacy-protected image retrieval scheme for fast and secure image search. Symmetry, 12(2): 282, 2020.

[110] Chengyuan Zhang, Lei Zhu, Shichao Zhang, and Weiren Yu. Tdhppir: an efficient deep hashing based privacy-preserving image retrieval method. Neurocomputing, 406:386–398, 2020.

[111] Ayad I Abdulsada and Nagham Abdulrasool Taha. Towards efficient privacy-

preserving image similarity detection. In AIP Conference Proceedings, volume 2144, page 050005. AIP Publishing LLC, 2019.

[112] Aiman Sultan, Shahzaib Tahir, Hasan Tahir, Tayyaba Anwer, Fawad Khan, Muttukrishnan Rajarajan, and Omer Rana. A novel image-based homomorphic approach for preserving the privacy of autonomous vehicles connected to the cloud. IEEE Transactions on Intelligent Transportation Systems, 2022.

[113] Peipeng Yu, Jian Tang, Zhihua Xia, Zhetao Li, and Jian Weng. A privacy-preserving jpeg image retrieval scheme using the local markov feature and bag-of-words model in cloud computing. IEEE Transactions on Cloud Computing, 2023.

[114] Constantin F Aliferis, Alexander Statnikov, Ioannis Tsamardinos, Subramani Mani, and Xenofon D Koutsoukos. Local causal and markov blanket induction for causal discovery and feature selection for classification part i: algorithms and empirical evaluation. Journal of Machine Learning Research, 11(1), 2010.

[115] Bin Li, Shilei Ding, and Xu Yang. A privacy-preserving scheme for jpeg image retrieval based on deep learning. In Journal of Physics: Conference Series, volume 1856, page 012007. IOP Publishing, 2021.

[116] Shahzaib Tahir, Sushmita Ruj, Yogachandran Rahulamathavan, Muttukrishnan Rajarajan, and Cornelius Glackin. A new secure and lightweight

searchable encryption scheme over encrypted cloud data. IEEE Transactions on Emerging Topics in Computing, 7(4):530–544, 2017.

[117] Yongkang Wong, Shaokang Chen, Sandra Mau, Conrad Sanderson, and Brian C. Lovell. Patch-based probabilistic image quality assessment for face selection and improved video-based face recognition. In IEEE Biometrics Workshop, Computer Vision and Pattern Recognition (CVPR) Workshops, pages 81–88. IEEE, June 2011.