A Dissertation

On

PRIVACY PRESERVING FACE SEARCH OVER CLOUD

Submitted in partial fulfillment of the requirements for the award of degree

of

Master of Technology

in

Computer Science and Engineering

Submitted By

RAJAT SHARMA

(16535033)

Under the guidance of

DR. BALASUBRAMANIAN RAMAN

Associate Professor, Dept. of Computer Science and Engineering

Department of Computer Science & Engineering

INDIAN INSTITUTE OF TECHNOLOGY, ROORKEE

Roorkee – 247667

May, 2018

AUTHOR'S DECLARATION

I declare that the work presented in this dissertation with title **"Privacy Preserving Face Search over Cloud"** towards fulfillment of the requirement for the award of the degree of **Master of Technology** in **Computer Science & Engineering** submitted in the **Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, India** is an authentic record of my own work carried out during the period of **May 2017** to **May 2018** under the supervision of **Dr. Balasubramanian Raman**, Associate Professor, Department of Computer Science and Engineering, Indian Institute of Technology Roorkee, Roorkee, India. The content of this dissertation has not been submitted by me for the award of any other degree of this or any other institute.

Date: .....
Place: ROORKEE

RAJAT SHARMA
(16535033)
M.TECH (CSE)

CERTIFICATE

This is to certify that the statement made by the candidate is correct to the best of my knowledge and belief.

Date: .....
Place: .....

Signature: .....
Dr. Balasubramanian Raman
(Associate Professor)
Indian Institute of Technology Roorkee

ACKNOWLEDGEMENTS

Dedicated to my family and friends, for standing by me through thick and thin, without whom I would not have gotten this far. I would like to express my sincere gratitude to my advisor **Dr. Balasubramanian Raman** for the continuous support of my study and research, for his patience, motivation, enthusiasm and immense knowledge. His guidance helped me in all time of research and writing of this thesis. I could not have imagined having a better advisor and mentor for my study. I would like to express my sincere appreciation and gratitude towards **Mr. Amitesh Singh Rajput** for his encouragement, consistent support and invaluable suggestions at the time I needed the most.

I am also grateful to the Department of Computer Science and Engineering, IIT Roorkee for providing valuable resources to aid my research.

RAJAT SHARMA

ABSTRACT

With the enormous amount of data, especially images, getting produced every minute and the ease of availability of cloud services at cheaper rates, there is an exponential increase in the amount of data being moved to the third-party cloud data centers. However, these data centers and the transmission of data are vulnerable to various intruding attacks resulting in untoward breaching of security and privacy. To avoid this, the visual information of image data is encrypted before transmission and storage. Albeit, performing image processing operations on these encrypted images is a challenging task, but at the same time, it saves computational and transmission costs as compared to computing those operations in other domains. This report investigates and explores privacy preserving face search over cloud. Particularly, given a face image as query, we attempt to find its region among encrypted images. We propose a novel privacy preserving face search model that extracts features from face image and identifies the face object region confined in the encrypted image.

Keywords: Cloud computing, privacy preserving, face search, image encryption.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

1.1 Cloud Computing –

One of the important services of Cloud Computing is Cloud Storage, through which data is remotely managed, maintained, and backed up, at the same time providing facilities like availability, accessibility and recoverability. The service allows different users to store files online, so that they can be accessed from any location using the Internet. It is now common for users to move their data from local devices to ubiquitous servers of the cloud to exploit its flexible and economic services. Moreover, to handle the massive volumes of data in image processing and utilization of huge data space availability, it is preferred to execute the computations of image processing on cloud servers [1].

1.2 Privacy Preserving Processing –

In the recent years, the swift technological developments in the areas like social networking, internet applications, clouds computing, etc. have raised vital concerns related to the security and privacy of user-related data. With the advent and rapidly increasing popularity of social media, privacy-related incidents and harms are increasing significantly. One of the major privacy concerns exists when the outsourced image data may leak secret information of the user, like personal identity, locality, or economic profiles. Also, these extracted features from the data may reveal important private and personal information [2].

Thus, it has now become imperative to include privacy preserving techniques to the processing of user related data. Also, the sensitive information is prone to vulnerabilities during the communication and handling at the cloud servers [6]. This has led to an inevitable need for implementing cryptographic techniques before the transmission and processing of data. One of the straight-forward ways of safeguarding a digital image is to encrypt the data beforehand.

Image processing in encrypted domain refers to the act of performing image processing operations on an encrypted image, so as to generate the desired results without decrypting and then performing the same operations on the actual image. It involves processing images while

upholding the security, integrity and privacy of the data. Hence, researching privacy preserving image processing over encrypted domain is of significant importance. In this report, we present a privacy preserving face search model which detects a face region in an encrypted image over a robust image encryption system. The results show significant performance of the presented model in terms of both, search efficiency and strength of encryption.

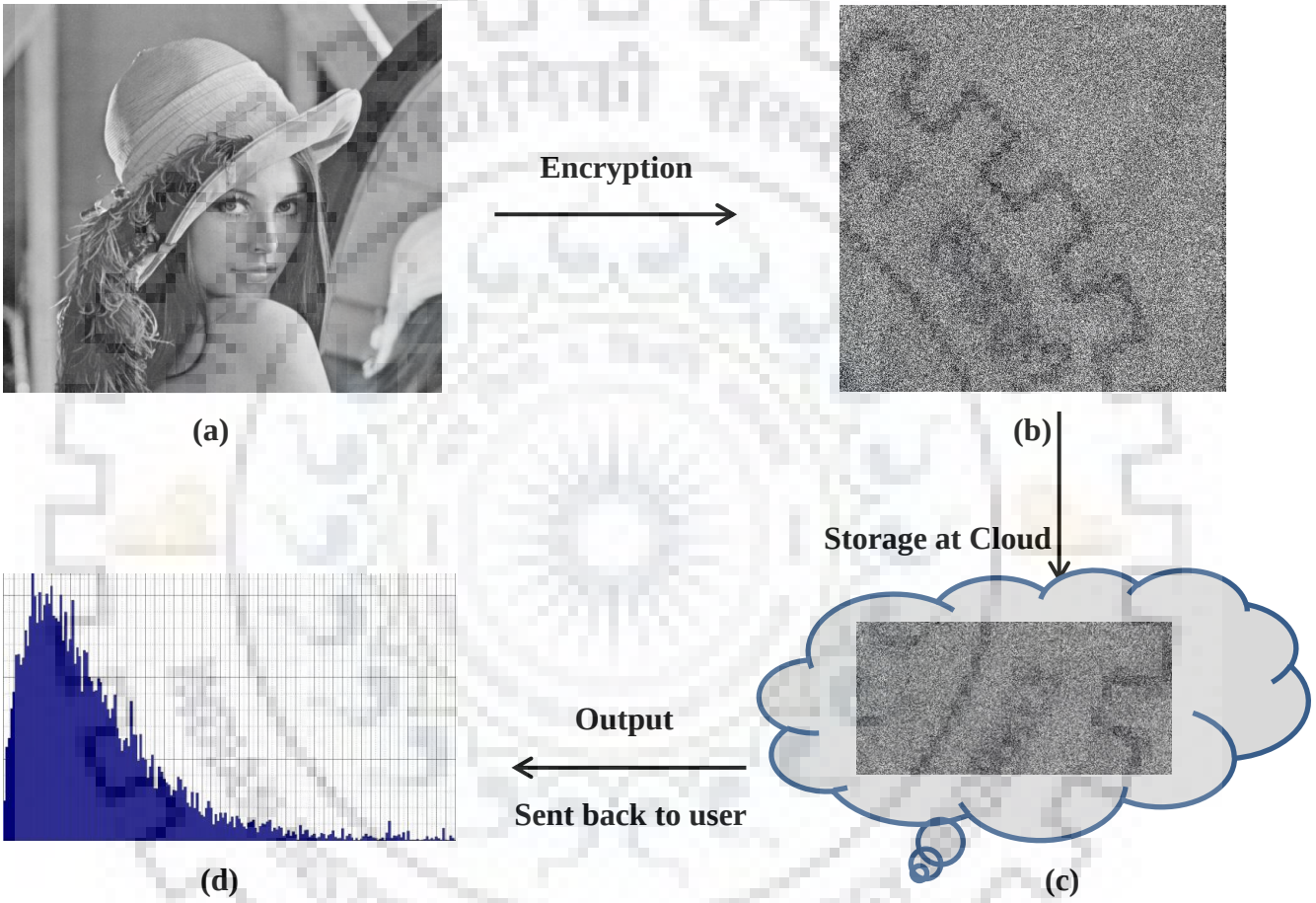


Figure 1.1 Image Processing in Encrypted Domain

Figure 1.1 shows an overview of image processing in encrypted domain. Figure (a) shows the original image. Figure (b) shows the image obtained after encryption which can now be stored at a cloud datacenter, shown in Figure (c). After performing the required operations over cloud, the desired output can be sent back to user, as shown in Figure (d).

2. LITERATURE REVIEW

2.1 *Recent Works in Encrypted Domain* –

Recent works in the field of encrypted domain are motivated towards processing an encrypted image in various aspects like face search [1], authentication systems [2], scaling and cropping images [3], extracting/detecting image features (SIFT) [4], video editing [5], etc. The authors in [3] proposed a model that facilitates the data servers to perform scaling and cropping operations in encrypted images. Thereby, the end users can subsequently access the required scaled or cropped region of the image from resultant encrypted images. So only the exact part of the encrypted images is sent to the end users, at the resolution requested by them.

While, in [4] the authors describe about the need to perform SIFT feature detection on images with high efficiency over cloud servers. To address this challenging task, the authors suggest ways for outsourcing the operation of SIFT computation and propose a novel approach of SecSIFT, a highly efficient privacy upholding SIFT feature extraction system. The authors presented a privacy preserving video editing model in [5] which performs the video processing operations over secure cloud servers. The model consisted of a secure video scaling and cropping scheme, based on Shamir's secret sharing, which allowed datacenters to execute scaling and cropping over encrypted videos without discovering their actual content.

2.2 *Face Detection in Encrypted Image* –

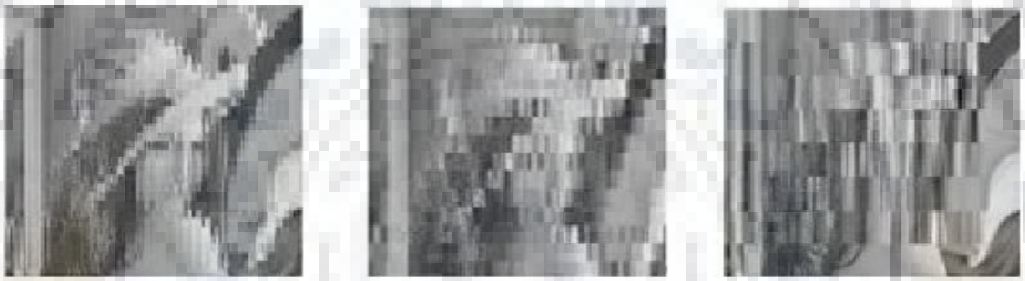
WQ Yan and MS Kankanhalli proposed a technique for face search in encrypted domain [1]. Their goal was to attempt face search on the encrypted images, i.e., provided a face image as the query image, to find it among images stored in encrypted form. This is performed by extracting features of given image and identifying the face object area in given encrypted image. The accuracy of the search experiments is then measured by computing and evaluating precision, recall & the F-measure.

The authors in [2] describe a privacy preserving face identification method using cryptographic methods. A single key is used for encryption of a single pixel value in given image input. Similarly, bitwise XOR calculation is done on every pixel of the input image with its

corresponding previously generated key. The resultant image obtained is then used to search faces, wherein eigenvalues are computed for Principle Component Analysis based search.

2.3 *Image Encryption* –

For the process of image encryption in [1], images are imported in spatial domain and then are segmented into identical sized blocks. A corresponding Hilbert Space-filling curve (HSC) is recursively generated for each block using a generator. After generation of the HSC curve, pixel sequence is sorted by following the pixel precedence on the generated HSC curve, resulting in a scrambled image. Figure 2.1 demonstrates the results of encrypting *Lena* image using HSC with different block sizes used for scrambling.



(a) **(b)** **(c)**

Figure 2.1 Scrambled *Lena* images in (a) 16 × 16, (b) 32 × 32 and (c) 64 × 64 block sizes using HSC [1]

It is evident that *Lena*'s face is becoming difficult to be identified from left to right. This HSC based image encryption modifies the pixel order of the image, but it does not modify color information of pixels.

Authors of [8] present a method which provides an efficient approach to the problem of image encryption by generating various 2D accessing algorithmic scan patterns of an image of $n \times n$ elements. There are two notions behind the specification of this approach: (i) the concept of hierarchical segmentation of an image into blocks or square regions, (ii) the concept of encryption of these blocks by performing a defined pixel scan over the block and regenerating them by sorting the pixels by the order in which they were traversed.

2.4 *Feature Selection and Matching* –

Feature selection and matching techniques are the essential steps of any face detection and recognition procedure. In [1], the images were encrypted based on DCT domain, and hence the features that were to be used for searching must remain invariant in both encrypted and DCT domain. Due to this, in [1], the authors proposed to use features like mean, variance and histogram for the face search. These statistical features are then combined in a vector and then hierarchical multi-resolution matching based on these vectors is performed by calculating the distances between the vectors.

While in [2], for performing the face search on the encrypted face images, an approach based on PCA is used. It uses eigenvalues of the face images. Eigenvectors are obtained from the encrypted face image. These eigenvectors are then used for training and testing process for the experiment.

2.5 *Research Gap* –

The face search method proposed by Wei Qi Yan and Mohan S Kankanhalli [1] scans over an encrypted image and executes the feature extraction and matching steps for searching the face object. The method uses HSC based encryption techniques for scrambling which follows a fixed pattern for traversing the image pixels. However, the work done in [12] describes about the cryptanalysis of HSC based encryption and hence making it a weaker choice for upholding the security, thus, the strength of shuffling based encryption needs to be improved. Further, the scrambled image can be encrypted by generating its secret shares and distributing them over different storage servers. Also, with the use of different set of feature vectors and metrics for distance calculation, there is room for further enhancement in search accuracy. Hence, with these shortcomings in [1], there is a scope of further improvement in terms of strength of encryption and accuracy of search and motivated us to devise an efficient method for searching face regions in encrypted images.

3. PROBLEM FORMULATION

This report deals with the challenge of accurately finding a face object confined in an encrypted image. Our problem is inspired by the face detection and recognition in the field of computer vision and search and retrieval in encrypted domain, while taking benefits of direct manipulation of encrypted data [1]. The multistage task of face region search in an encrypted image can be divided into three parts – **Image Encryption, Feature Selection,** and **Feature Matching**.

Image Encryption refers to the concept of taking the pixel bits of an image and collectively rearranging or modifying their values using a defined logic [1], thereby leading to a completely new set of pixels, which is different from the original combination and thus, ensuing in obscuring visual information of the image. This way of encryption is unlike the conventional encryption algorithms like ECC, AES, etc. which increase the computations while deploying over images. The advantage of using such techniques for encryption is that they disturb the original sequence of pixel positions of the image, and hence the pixel neighborhood operations like SIFT, detection of edges, etc., cannot be achieved anymore. However, at times, these geometric information can be extracted from images which were ciphered using ECC or AES techniques [1], which is not desirable.

Typically, a face-search algorithm in spatial domain involves extracting facial features describing a face object image and then searching is done by comparing selected facial features from the image and a face database. Similarly, the step of feature selection and extraction is performed over images in encrypted domain, however, the set of features used are usually different from the ones used in spatial domain. This is due to the fact that once an image undergoes an encryption or scrambling of pixels, the facial features are lost in the process and hence cannot be used for matching. So, we need to look for features and properties of the face object which remain invariant in both, the original image and its encrypted version. These features are usually statistical properties of an image like mean, variance, histogram, etc.

Once the feature selection is done, the final step is to search for the target face region. Since the facial region can lie in any part of the complete image and at unknown resolution, we need to

match features by sliding and moving over the whole image in hierarchical fashion at multiple resolutions in order to find its location. Thus we need to traverse the input image in scan-line order by moving left to right and top to bottom, and match the features by computing the distances between the vectors formed by combining the extracted features of the regions of the encrypted image and face image. Once the bottom-right end of the image is reached, we alter the resolution and compute the distances by beginning from topmost left of the image again, until significant resolutions of scrambled face image have been scanned, i.e., at multiple resolutions.

4. PROPOSED APPROACH

In this segment, we propose the approach we will follow to deal with the problem of detecting a face region in an encrypted image. In our proposed approach, the process of image encryption is performed on an intermediary **Trust Server** (TS) and the process of face detection by feature matching is performed on the available **Cloud Servers** (CS). The detailed description of the overall architecture is presented in section 4.2.

4.1 *Preliminaries* –

4.1.1 *Image Encryption* –

For encrypting images using scrambling, rather than using HSC based approach for shuffling pixel positions as used in [1], we propose a two-stage encryption system; **(i) scrambling using scan patterns** [8] and **(ii) secret share generation** [11]. In the first stage, as discussed earlier in section 2.3, the scan patterns perform image encryption by generating various 2D accessing algorithmic scan patterns of an image constituting $\boldsymbol{n \times n}$ elements. The first step is to divide the image into blocks of size $\boldsymbol{t \times t}$ (we have considered 32×32 sized blocks for the implementation purpose). Then, each of these blocks are traversed by performing a defined pixel scan over the block and regenerating them by sorting and arranging the pixels by the order in which they were traversed. We segment the image into blocks because we prefer that even after scrambling, the pixel should still remain in the neighborhood of its actual position.

Advantage of using scan patterns is the availability of different variety of traversing patterns, which if complemented by a key based encryption algorithm for each block, can lead to highly secure and robust encrypted image. So, scanning the image provides an encryption of it, i.e., a transposition transformation of the 2D image data. The authors in [8] propose various orders or traversing patterns, different than the traditional raster scan, which can be used to scan the image and scramble it to obscure the visual data. Figure 4.1 demonstrates scan patterns which can be used to traverse an image of $n \times n$ elements. Figure 4.2 shows the results of traversing and reordering the pixels over *Lena* image using the corresponding scan patterns shown in figure 4.1.

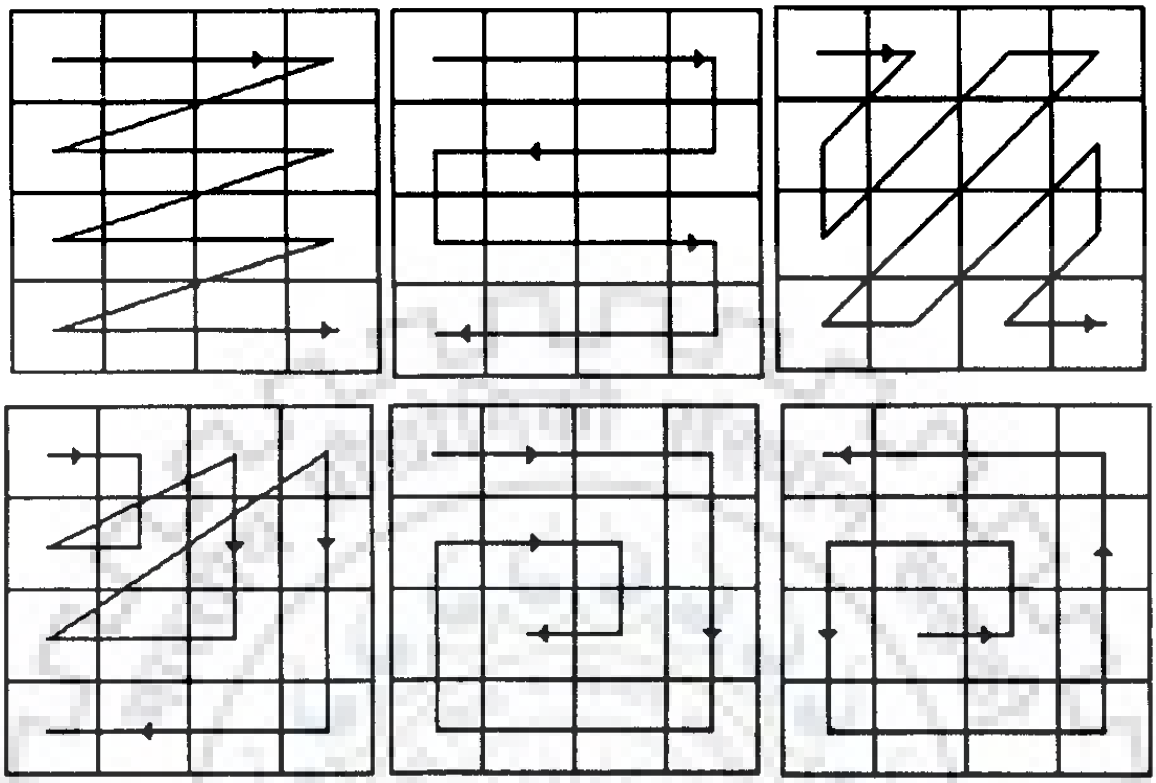


Figure 4.1 Various scan patterns for image of $n \times n$ elements [8]

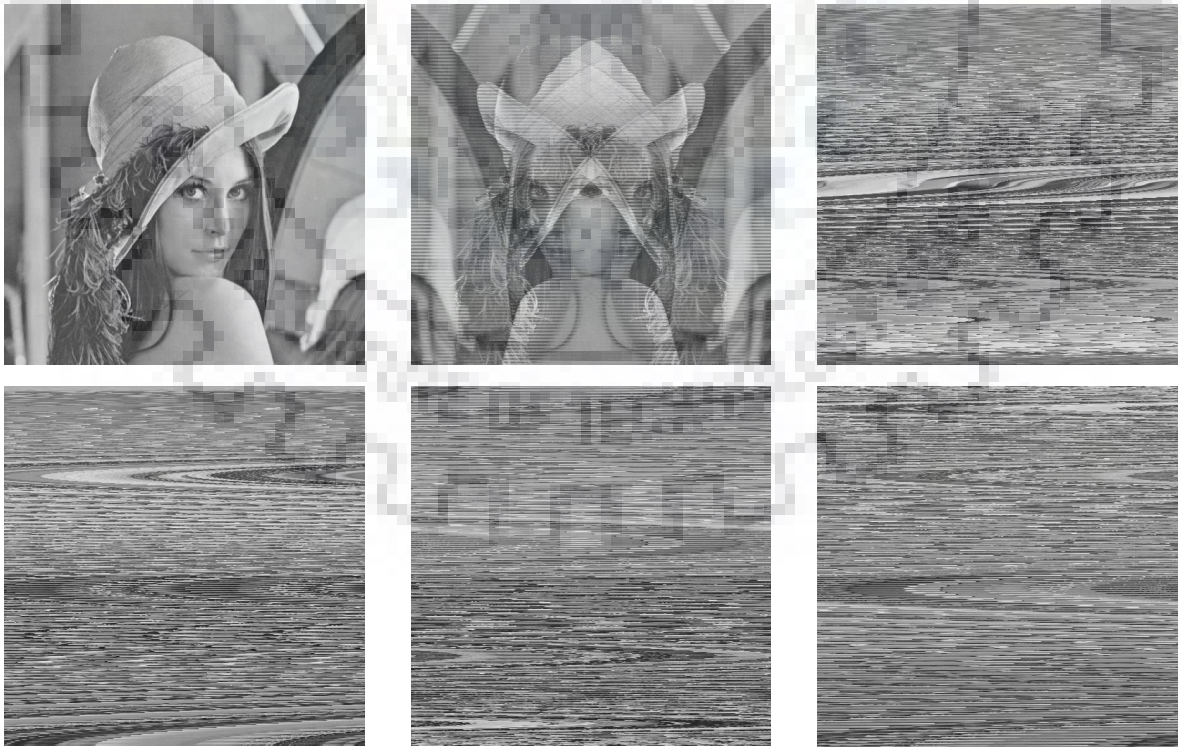


Figure 4.2 Scrambled *Lena* images using corresponding scan patterns shown in figure 4.1

In the second stage for encryption, we implement the model of secret sharing to obtain stronger and robust encryption, both visually and computationally. Secret sharing methods split any information into multiple shares in a way where any single share by itself does not possess any meaningful data, but altogether, they preserve the originality of the information, in whole.

For our purposes, we propose the use of a modulo based transformation over scrambled image, which generates each image share by computing $v \ mod \ p_i$ with different prime numbers (p_i) for different shares, where v is a pixel in the scrambled image.

To understand the process, consider the processing of a scrambled image received after first stage of encryption. Since every pixel value is handled separately, consider a pixel with intensity equal to 70. Now, if the given image is to be disintegrated into 3 image shares with prime numbers 23, 29, and 37, then, the corresponding pixel intensity values in the generated shares would be 1, 12 and 33 respectively.

4.1.2 *Feature Selection and Matching* –

The feature selection and matching is the most vital step of any object detection model. Here, since the visual features of image get destroyed by the scrambling step, we look for the features which remain same before and after the scrambling of image. Our proposed set of features to be used for searching includes features like **Mean, Standard Deviation** and **Entropy**. These features are immune against the scrambling process on the image and remain invariant in case of rotating, scaling and flipping as well. Thus, these features suit best in our interests and hence can be used for the search process. These features can then be combined to form a vector for feature vector matching between search images and query image. The eq. (1) and (2) describe the general form of respective feature vectors for search image share and query image share.

$$\mathbf{V_S = [F_{S1}\, F_{S2}\, F_{S3} \ldots\ldots\ldots F_{Sm}]} \qquad (1)$$

$$\mathbf{V_Q = [F_{Q1}\, F_{Q2}\, F_{Q3} \ldots\ldots\ldots F_{Qm}]} \qquad (2)$$

Since the facial region can lie in any part on an image and at unknown resolution, we propose the use of a sliding window, which moves over the complete search image, hierarchically, at

multiple resolutions. Each feature vector is normalized before performing any distance calculation.

We propose a two-step approach for matching the features. We call them **Level-1** matching and **Level-2** matching. In Level-1 matching, we compute the distance across the vectors of query image share and sliding window of search image share. This distance calculation does a low level similarity between the vectors. So, we compute pairwise Euclidean distance of feature vectors (mean, entropy, standard deviation) of query image share and each instance of sliding window in search image share, say *d1*, *d2* and *d3*, respectively for each feature. Before moving to the second step, we reduce the search space by removing the results obtained from Level-1 matching by a certain empirical threshold. So, we compute and store the product of the distances (*d1*, *d2* and *d3*) and prune results by selecting lowest 25% values (empirical threshold). And then finally perform the Level-2 match over the reduced search space, by performing a rigorous distance calculation between the vectors to search the face image. Now, in the second step, we compute and store the sum of the distances (*d1*, *d2* and *d3*) and perform Level-2 matching over the pruned data of Level-1. In the end, we output back the instance of sliding window with least value obtained in the Level-2 match. Figure 4.3 describes the flowchart of operations performed over cloud server for face region detection.

4.2 *Overall Architecture* –

The proposed approach is designed to run in cloud environment. The architecture we propose is based on "**honest-but-curious adversary model**", wherein the **Cloud Servers (CS)** execute the required tasks, however are eager to know about the content of the data. Therefore, we initially encrypt user images at an intermediary **Trust Server** (**TS**) before transmitting data to the cloud servers. Whenever required, the user sends images to the trust server for storage which further performs the task of two-staged encryption and generates *n* shares of the image. Each share is then transmitted to a different cloud server, which stores the share received. Upon receiving a request for face region detection along with a query image from user, the trust server generates the shares of the query image in the similar fashion and transmits those query image shares to the corresponding cloud servers. Each cloud server performs feature extraction and matching steps, as discussed in section 4.1.2, over the stored image share and returns a face region to the trust

server. The trust server then computes the position of the facial region based on the individual positions received from each of the *n* cloud servers and transmits the resulting face region back to the user. Whenever required, the trust server can also perform decryption on assembled shares, to regenerate the original image by following the reverse steps of encryption. Figure 4.4 shows the overall architecture of the proposed model.

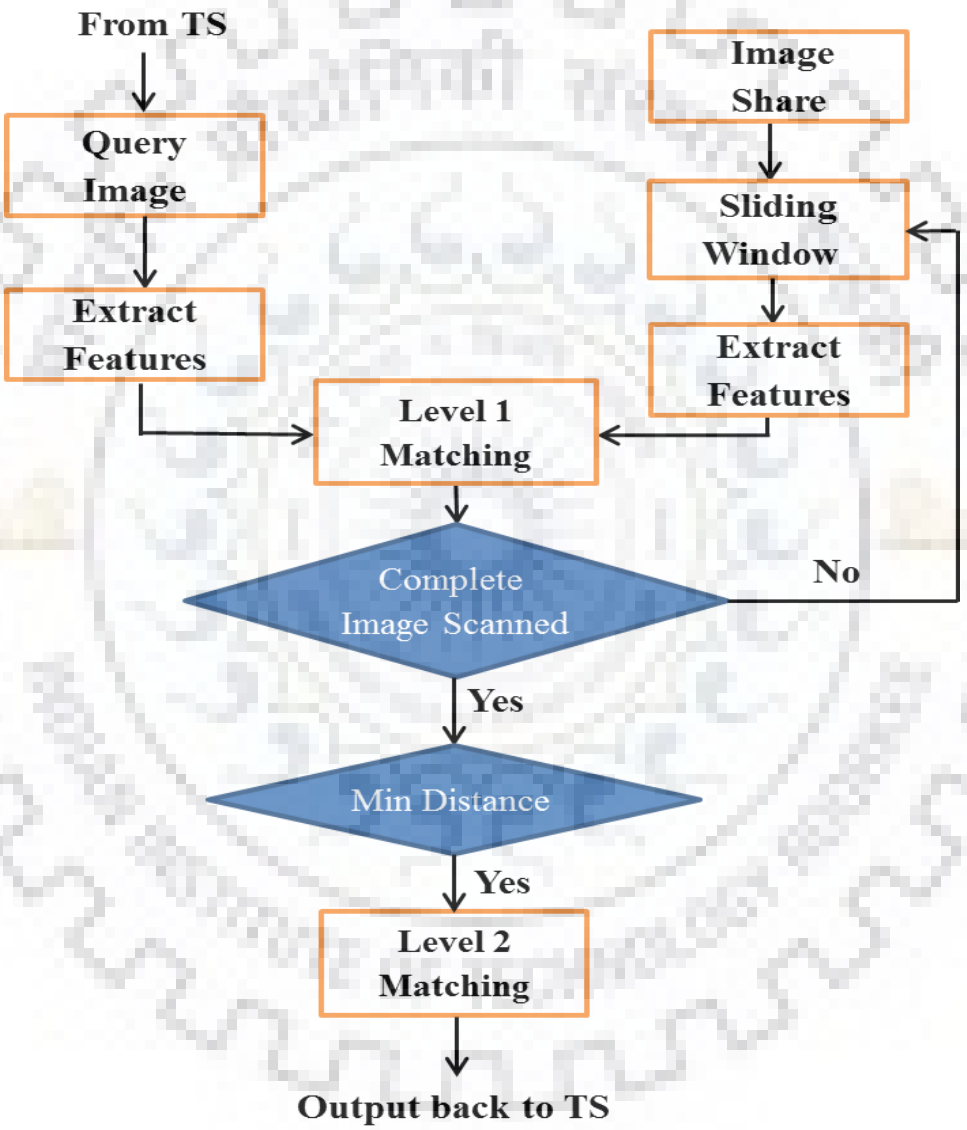


Figure 4.3 Flowchart of operations performed over Cloud Server for Face Region Detection

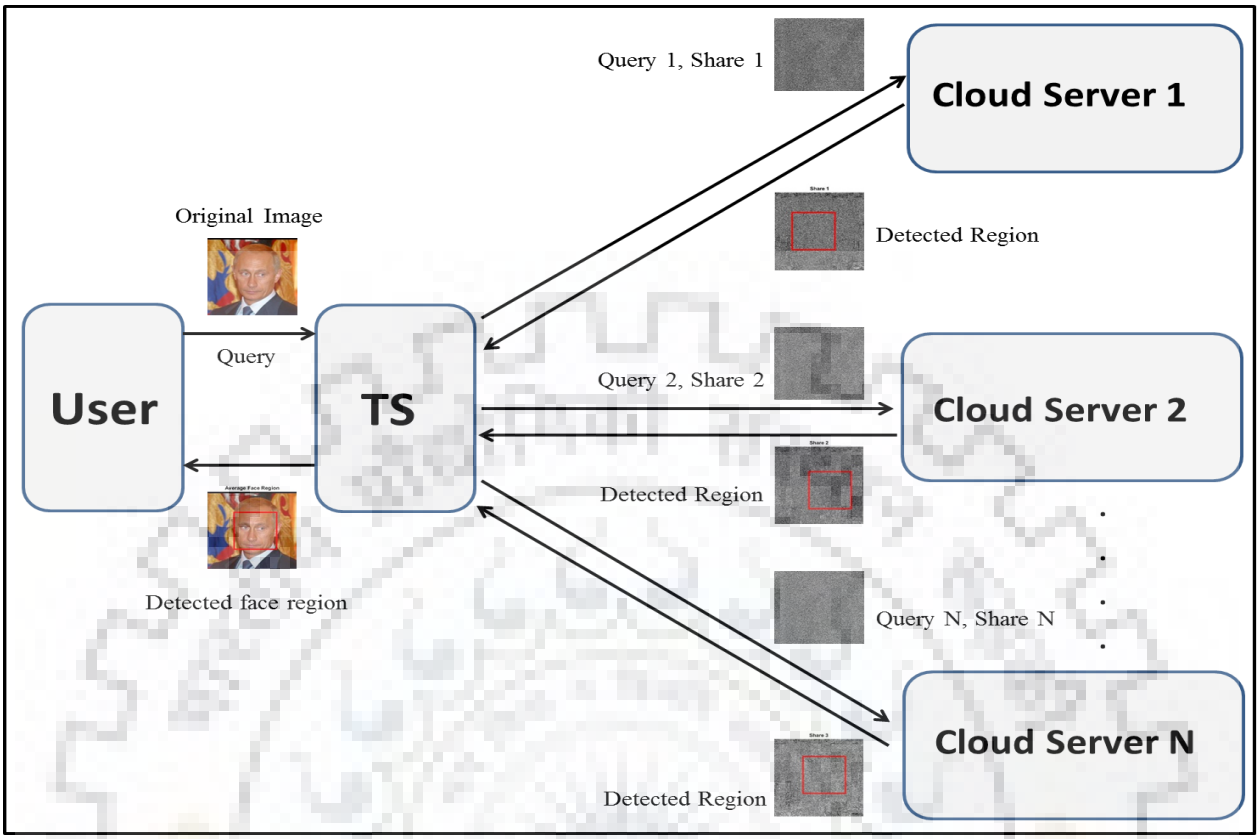


Figure 4.4 Overall architecture of the proposed approach

4.3 Proposed Methodology –

4.3.1 User to Trust Sever (TS):

1. User sends original image to TS.
2. Whenever needed, user sends a request along with a query image to TS for face region detection.

4.3.2 Processing at TS:

1. Segment the original image into blocks and perform scrambling on each block using a key to obtain a scrambled image. Generate *n* different shares of the encrypted image by taking modulo of each pixel value with *n* different prime numbers.
2. Transmit each of the *n* generated shares to *n* different **Cloud Servers** (CS).
3. Encrypt and generate *n* shares for query image in the same way, whenever a request from user is received.
4. Transmit each of the *n* generated shares of query image to corresponding cloud servers.

4.3.3 Processing at each CS:

1. Extract features (Mean, Entropy and Standard Deviation) of query share.
2. Run a sliding window at a given resolution over the face image share and extract features (Mean, Entropy, Standard Deviation).
3. Compute pairwise Euclidean distance of feature vectors of query image and each instance of sliding window. Store the product of the distances and perform Level-1 matching by selecting lowest 25% values (empirical threshold).
4. Compute and store the addition of the distances and perform Level-2 matching over the pruned data of Level-1.
5. Transmit the instance of sliding window with the least distance in the Level-2 matching, back to the TS.

4.3.4 Result generation at TS:

1. Receive the frames from each CS.
2. Compute the position of facial region by averaging the positions of each frame received.
3. Transmit the position of detected face region back to user.
4. Decrypt the image using CRT and following the encryption steps in reverse order.

5. EXPERIMENTATION AND RESULTS

5.1 Dataset –

The dataset used for experiments in the attempt to detect face region in encrypted images is "**Labeled Faces in the Wild** (**LFW**) Dataset" [10]. This dataset consists of more than 13,000 images with single face objects and is widely used in many face detection and recognition related works in the spatial domain.

5.2 Search Evaluations –

Now we will discuss how we assess our face detection results. Let us assume a face region has been identified in an image from a given set of numerous encrypted images. To evaluate the efficiency of our face region detection approach in encrypted images, we need to calculate statistical measures of the results such as, true positive *tp*, false positive *fp*, true negative *tn*, and false negative *fn*. Once we have our search results, we can calculate the recall, precision and F-measure values using the ground truth to assess the performance. The results show whether we have accurately detected the face region or not. We calculate,

$$\mathrm{Pr} = \frac{\mathrm{Tp}}{\mathrm{Tp} + \mathrm{Fp}}$$

$$\mathrm{Rc} = \frac{\mathrm{Tp}}{\mathrm{Tp} + \mathrm{Fn}}$$

where Pr stands for precision and Rc for recall. Tp, Tn, Fp, and Fn are the true positive, true negative, false positive and false negative values found in the search. These values describe how precisely the results indicate the ground truth. Further, we calculate F-measure by,

$$\mathrm{Fm} = \frac{2 * \mathrm{Pr}.\mathrm{Rc}}{\mathrm{Pr} + \mathrm{Rc}}$$

5.3 Evaluation Results –

We implemented our face detection model over LFW dataset using MATLAB platform. We shattered both, the search image and the query image, into secret shares using the proposed encryption mechanism and performed feature extraction and matching, as discussed in chapter 4.

Face Data	Precision (Kankanhalli et. al) [1]	Precision (Proposed Approach)
Putin	0.825	**0.86**
Agassi	0.722	**0.80**
Clinton	0.690	**0.84**

Table 5.1 Comparison of Precision values between [1] and proposed approach

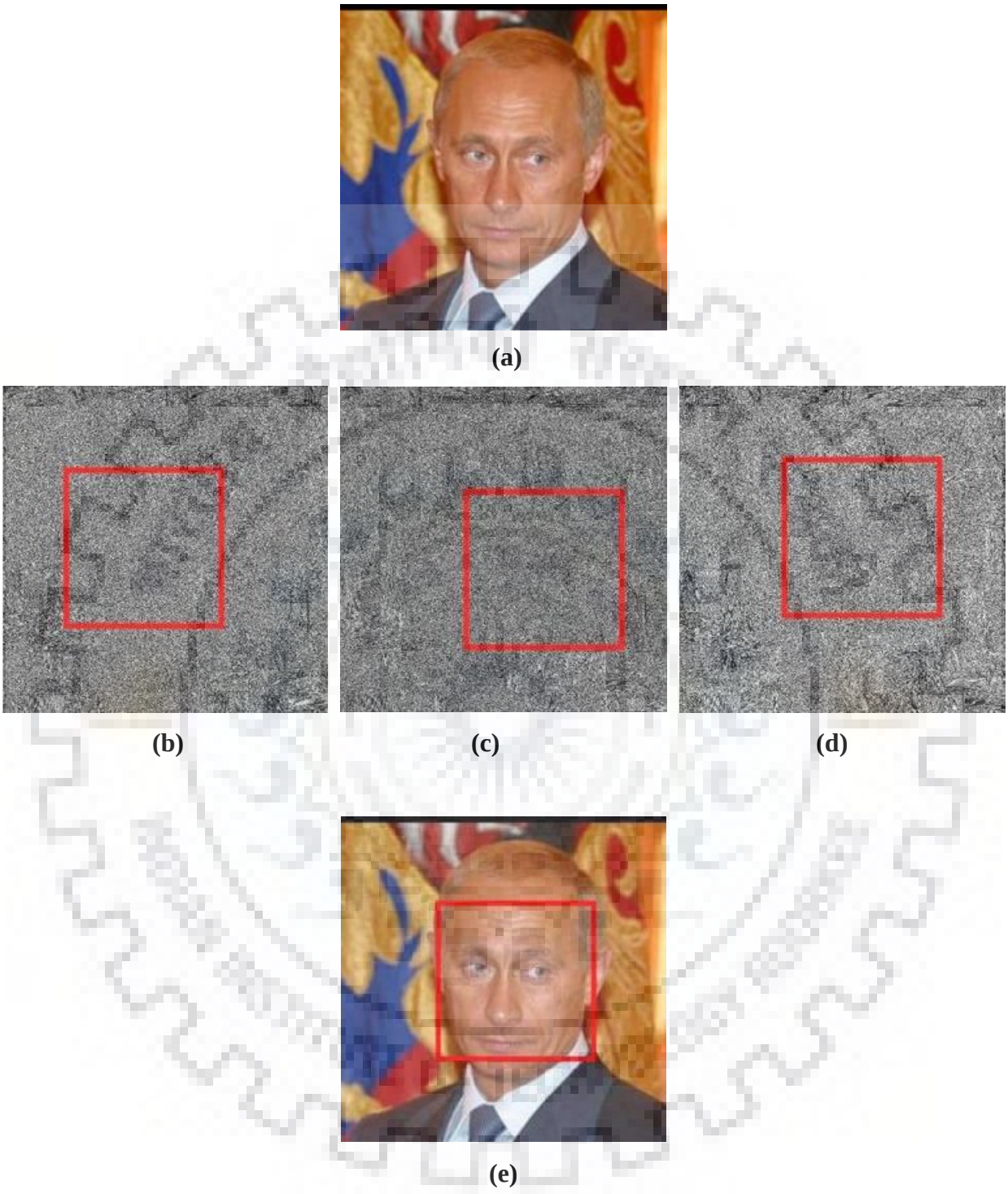
Face Data	Recall (Kankanhalli et. al) [1]	Recall (Proposed Approach)
Putin	0.354	**0.48**
Agassi	0.292	**0.45**
Clinton	0.202	**0.45**

Table 5.2 Comparison of Recall values between [1] and proposed approach

Face Data	F-Measure (Kankanhalli et. al) [1]	F-Measure (Proposed Approach)
Putin	0.496	**0.61**
Agassi	0.462	**0.57**
Clinton	0.312	**0.59**

Table 5.3 Comparison of F-Measure values between [1] and proposed approach

Table 5.1, Table 5.2 and Table 5.3 show the comparison of respective values of precision, recall and F-measure obtained in the implementation of the model used in [1] and the model proposed in this report, over a subset of LFW dataset. It is evident from the above tables that the performance of the proposed approach outperforms the results from [1] in terms of efficiency of face search.



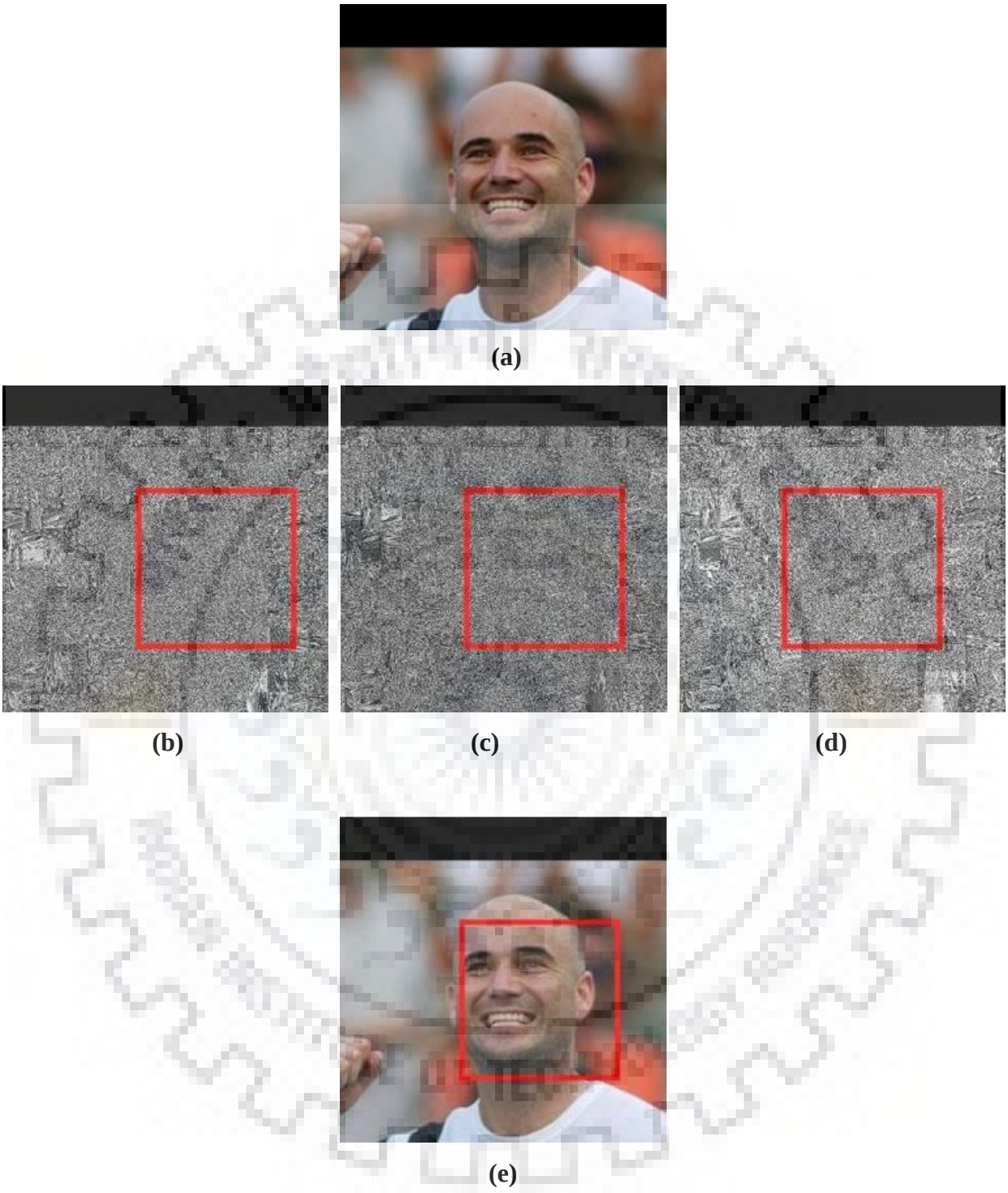
(a)

(b) **(c)** **(d)**

(e)

Figure 5.1 Face Region Detection (Putin)

Figure 5.1 (a) shows an original image of Vladimir Putin from LFW dataset [10]. Figures (a), (b) and (c) show the highlighted region of the face region detected in respective image shares. Figure (e) shows the averaged face region sent back to user as output.



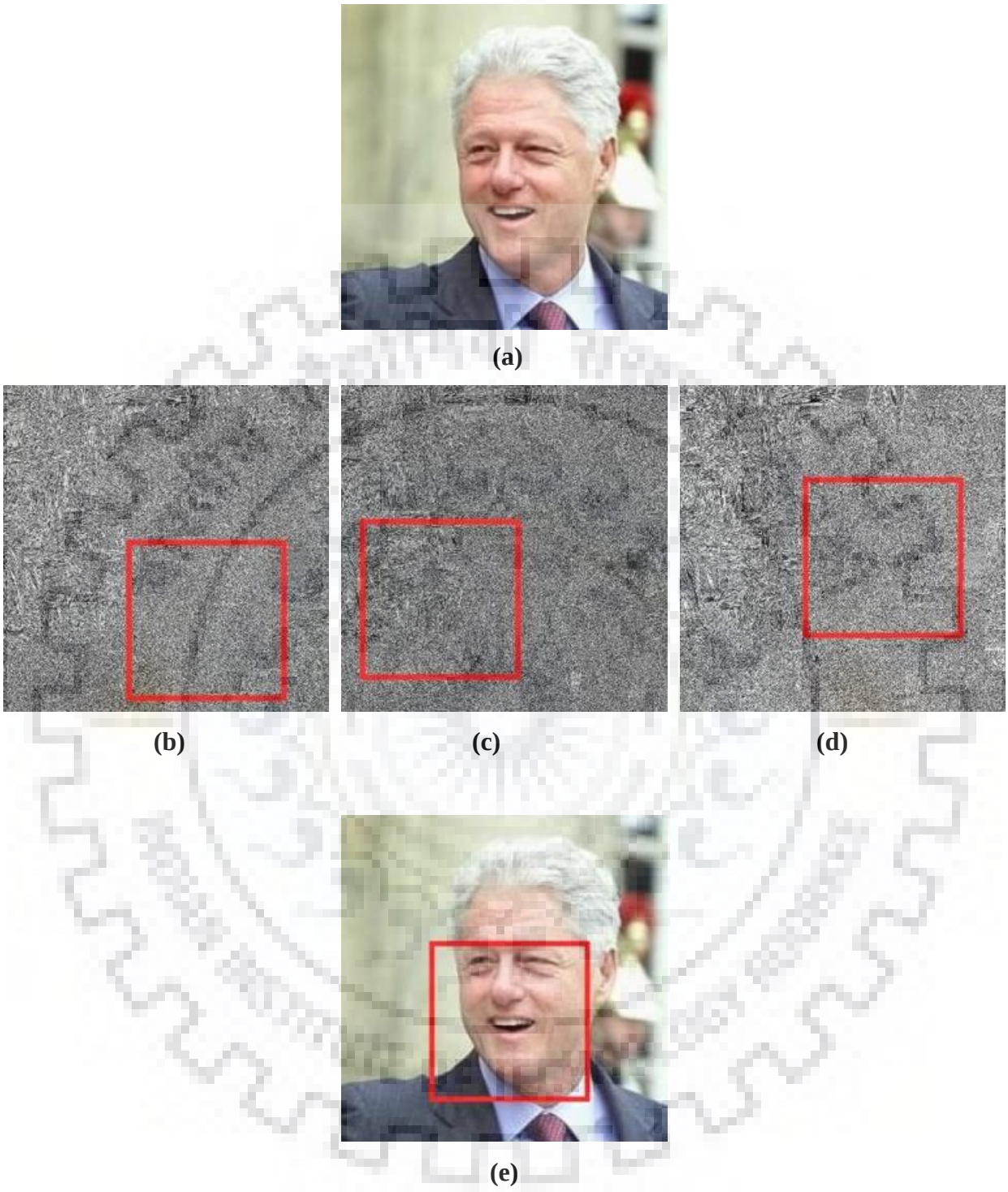
(a)

(b) **(c)** **(d)**

(e)

Figure 5.2 Face Region Detection (Agassi)

Figure 5.2 (a) shows an original image of Andre Agassi from LFW dataset [10]. Figures (a), (b) and (c) show the highlighted region of the face region detected in respective image shares. Figure (e) shows the averaged face region sent back to user as output.



(a)

(b) **(c)** **(d)**

(e)

Figure 5.3 Face Region Detection (Clinton)

Figure 5.3 (a) shows an original image of Bill Clinton from LFW dataset [10]. Figures (a), (b) and (c) show the highlighted region of the face region detected in respective image shares. Figure (e) shows the averaged face region sent back to user as output.

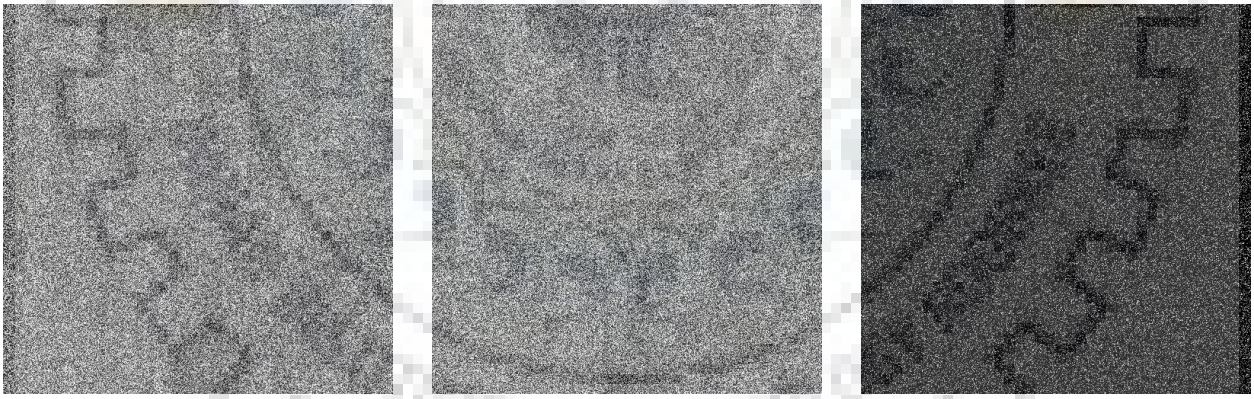
5.4 Security Analysis –

5.4.1 Key Space –

Secret key space is defined as the number of keys that can be used during encryption. A brute-force attack is one which involves methodically inspecting through all the keys possible in the space until we find the correct one. The key space must be broad enough to outclass such attacks. We have used *4*n* bit key for encryption of images, where *n* is the number of blocks generated after initial segmentation of original image and hence our key space is 2^{4n}, which is significantly vast.

5.4.2 Key Sensitivity –

In case of image encryption, key sensitivity is defined as change rate of number of pixels of the encrypted image when only one bit of the key is changed. Figure 5.4 (a) shows the generated *Lena* image share encrypted using key *k1* and (b) using key *k2*, with only one bit different from K1, while (c) shows the difference between the image shares (a) and (b). It can be observed from (c) that still, no image information is revealed from the difference of (a) and (b) shares.



(a) **(b)** **(c)**

Figure 5.4 Key Sensitivity

5.4.3 Structural Similarity (SSIM) –

SSIM is used for measuring the structural similarity between the two images by calculating the perceived change in structural information among the images. Table 5.4 shows the structural similarity (SSIM) index of the scrambled image and its shares with respect to the original image.

In case of image encryption, lower the value of SSIM, more is the strength of encryption and the lower values of SSIM in table manifest strength of our encryption system.

Image	Scrambled	Share 1	Share 2	Share 3	Avg. of Shares
Lena	0.2741	0.0124	0.0151	0.0103	**0.0126**
Mona Lisa	0.2978	0.0117	0.0141	0.0094	**0.0117**
Cameraman	0.4553	0.0110	0.0122	0.0096	**0.0109**

Table 5.4 SSIM values of scrambled image and its shares with original image

5.4.4 Peak Signal-to-Noise Ratio (PSNR) –

Table 5.5 demonstrates the PSNR between the scrambled image and its shares with the original image. Higher the value of PSNR, closer the encrypted image is to its original image. Hence, for better encryption system, the PSNR values should be as low as possible. It can be observed from table that the PSNR values of shares are quite low.

Image	Scrambled	Share 1	Share 2	Share 3	Avg. of Shares
Lena	15.9562	10.0251	10.0344	9.1957	**9.7517**
Mona Lisa	19.4712	10.0013	10.3622	8.8461	**9.7365**
Cameraman	14.7232	9.1876	9.0654	8.7513	**9.0014**

Table 5.5 PSNR values of scrambled image and its shares with original image

5.4.5 Normalized Cross Correlation (NCC) –

NCC across two images measures how pixel at *(x, y)* of one image is correlated with the pixel at *(x, y)* of the other. Table 5.6 shows the NCC between the original image and the scrambled image and its shares for *Lena*, *Mona Lisa* and *Cameraman* images. The value of the NCC ranges from -1 to 1. If there is no distortion in the encrypted image, then the value of NCC will be 1. In case of image encryption, closer the value of NCC to 0, more is the strength of encryption.

Image	Scrambled	Share 1	Share 2	Share 3	Avg. of Shares
Lena	0.9534	0.8178	0.8050	0.8013	**0.8080**
Mona Lisa	0.9651	0.7853	0.7747	0.7721	**0.7773**
Cameraman	0.9384	0.7795	0.7593	0.7813	**0.7733**

Table 5.6 NCC values of scrambled image and its shares with original image

5.4.6 Correlation between Adjacent Pixels –

Correlation coefficients depict the linear relation between two variables (adjacent pixels in case of images) and range from -1 to 1. Correlation coefficients with value close to 0 (both positive and negative), relate to excellent encryption strength. Table 5.7, Table 5.8 and Table 5.9 show the Diagonal, Horizontal and Vertical Correlation coefficients, respectively, for scrambled image and its image shares for *Lena*, *Mona Lisa* and *Cameraman* images. It can be observed from table that the correlation values of shares are significantly low, hence, demonstrating the strength and robustness of our encryption model.

Image	Scrambled	Share 1	Share 2	Share 3	Avg. of Shares
Lena	0.6696	0.0044	0.0053	0.0117	**0.0071**
Mona Lisa	0.8083	5.17e-05	0.0209	0.0041	**0.0083**
Cameraman	0.7437	0.0030	0.0401	0.0237	**0.0222**

Table 5.7 Diagonal Correlation values of scrambled image and its shares

Image	Scrambled	Share 1	Share 2	Share 3	Avg. of Shares
Lena	0.8064	0.0053	0.0229	0.0191	**0.0157**
Mona Lisa	0.9270	0.0039	0.0028	0.0113	**0.006**
Cameraman	0.8447	0.0384	0.0491	0.0653	**0.0509**

Table 5.8 Horizontal Correlation values of scrambled image and its shares

Image	Scrambled	Share 1	Share 2	Share 3	Avg. of Shares
Lena	0.8142	0.0051	0.0140	0.0026	**0.0072**
Mona Lisa	0.9305	5.65e-04	3.41e-04	0.0157	**0.0055**
Cameraman	0.8479	0.0425	0.0525	0.0632	**0.0527**

Table 5.9 Vertical Correlation values of scrambled image and its shares

5.4.6 Histogram Analysis –

Figure 5.5 shows various elements of *Lena* image encryption. Figure (a) shows original *Lena* image. Figure (b) shows the result of first stage of the proposed encryption system, i.e., the scrambled version of original image using scan patterns. Figure (c) shows histogram of the original *Lena* image, while figures (d), (e) and (f) respectively show the three generated secret

shares during the second stage of encryption. Figures (g) (h) and (i) show the histograms of generated image shares with desired uniformity in the distribution of pixel intensities.

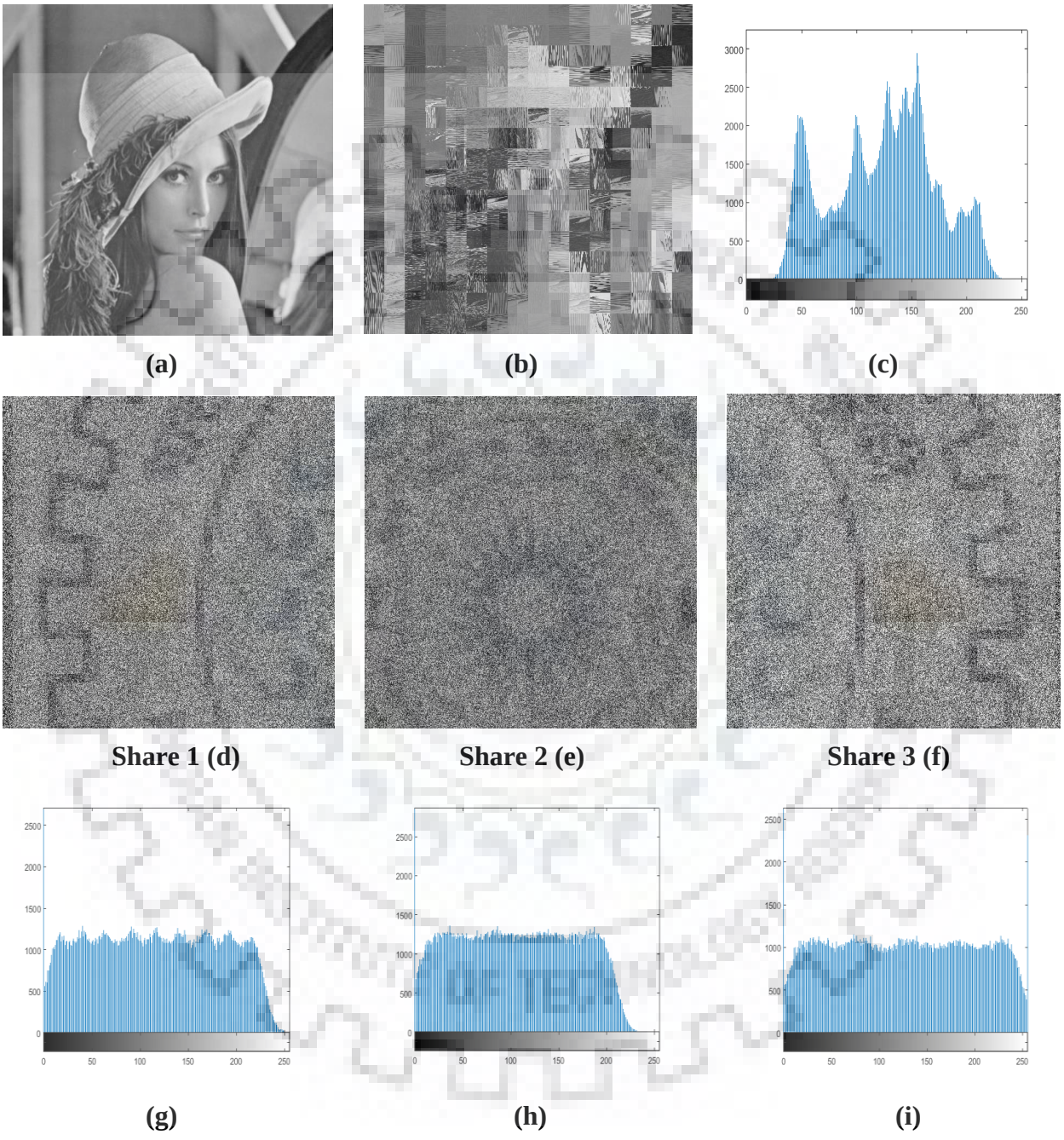


Figure 5.5 Histogram analysis of *Lena* and its shares

Figure 5.6 (a) shows original *Cameraman* image. Figure (b) shows the scrambled version of original image using scan patterns. Figure (c) shows histogram of the original *Cameraman* image, while figures (d), (e) and (f) respectively show the three generated secret shares during the second stage of encryption. Figures (g) (h) and (i) show the histograms of generated image shares with desired uniformity in the distribution of pixel intensities.

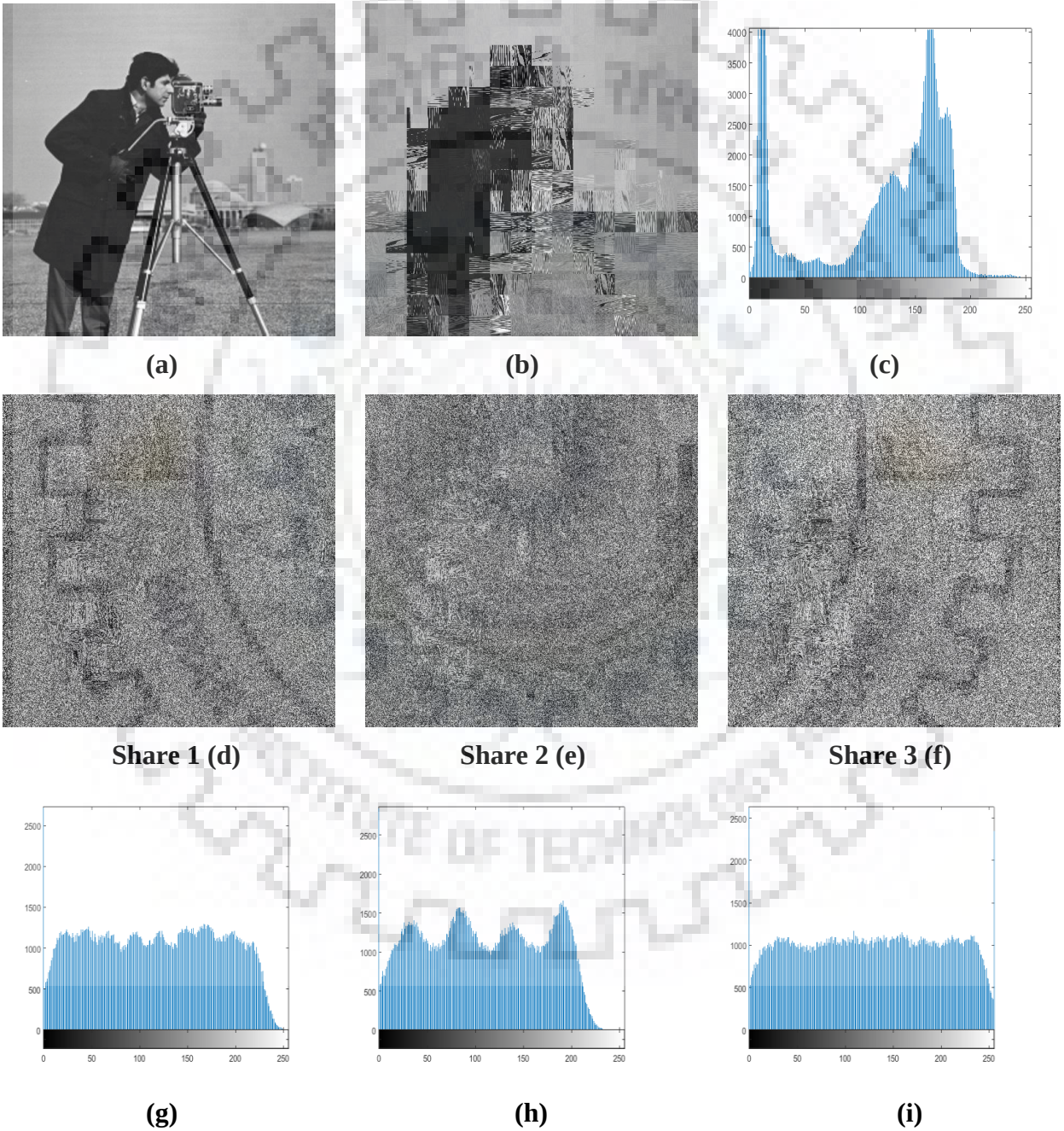


Figure 5.6 Histogram analysis of *Cameraman* and its shares

Figure 5.7 Precision of face detection in encrypted images using LFW dataset: 80% (Agassi)

Figure 5.8 Precision of face detection in encrypted images using LFW dataset: 84% (Clinton)

Figure 5.9 Precision of face detection in encrypted images using LFW dataset: 86% (Putin)

6. CONCLUSION AND FUTURE SCOPE

In this report, we presented a novel and efficient method for face detection in encrypted images. The sole purpose of using encrypted domain techniques is to use the encrypted data for processing to reduce the computation time and speeding-up the overall procedure by directly operating on the encrypted data, without learning about its actual content.

We discussed the recent works of face search in encrypted domain, image encryption, feature selection and matching and explored the challenges dealt. We proposed a novel approach for face search in encrypted domain and implemented it over the LFW dataset. This dataset has been a benchmark for many face detection and recognition related works globally. Our method of two-staged encryption using scan patterns and secret share generation, and feature set extraction and matching for face search in encrypted images over the LFW dataset shows significant improvement in results both, in terms of security and accuracy, compared to the previously used approach [1]. The proposed two-staged encryption mechanism provides sturdy and robust security against malicious attacks. More the number of shares generated, more is the security and efficiency of search. However, generating more shares will lead to higher computation and infrastructural cost. Moreover, as speculated, we also observed a trade-off between the accuracy of face region detection and the quality of encryption.

We believe that future scope and possibilities of encrypted domain processing are huge and they can be explored to greater extent for further improvements, not only in terms of encryption and efficiency, but in exploring different fields where privacy of data is a major concern as well. Moreover, the approach proposed in this report is not limited to only face-region detection in encrypted images, since it does not make use of any specific facial feature for region detection. Hence, the model can be further extended to perform secure and efficient privacy preserving object detection over encrypted data. Also, this concept can be further extended to perform face detection in videos using CCTV cameras, by fragmenting and periodically analyzing different frames of the video, to perform encrypted video surveillance. This encrypted video surveillance can assist in monitoring any high-profile area, without needing much human intervention, while preserving the privacy of the data at the same time.

7. REFERENCES

[1] Yan, Wei Qi, and Mohan S. Kankanhalli. Face search in encrypted domain. In, *Pacific-rim Symposium on Image and Video Technology*, pp. 775-790. Springer, Cham, 2015.

[2] Ergun, Ovgu Ozturk. Privacy preserving face recognition in encrypted domain. In, *IEEE Asia Pacific Conference on Circuits and Systems (APCCAS)*, pp. 643-646. IEEE, 2014.

[3] Mohanty, M., Ooi, W. T., & Atrey, P. K. Scale me, crop me, know me not: Supporting scaling and cropping in secret image sharing. In, *IEEE International Conference on Multimedia and Expo (ICME)*, (pp. 1-6). IEEE, 2013.

[4] Qin, Zhan, Jingbo Yan, Kui Ren, Chang Wen Chen, and Cong Wang. Towards efficient privacy-preserving image feature extraction in cloud computing. In Proceedings of the *22nd ACM international conference on Multimedia*, pp. 497-506. ACM, 2014.

[5] Mohanty, Manoranjan, and Pradeep K. Atrey. Don't See Me, Just Edit Me: Towards Secure Cloud-based Video Editing. *11th Annual Symposium on Information Assurance (ASIA),* Albany, NY, 2016.

[6] Kiya, Hitoshi, and Masaaki Fujiyoshi. Signal and image processing in the encrypted domain. *ECTI Transactions on Computer and Information Technology (ECTI-CIT)* 6, no. 1: 10-17, 2012.

[7] Gonzalez, R. C., and R. E. Woods. Digital image processing: Pearson prentice hall. *Upper Saddle River, NJ*, 2008.

[8] Bourbakis, N., and Christos Alexopoulos. Picture data encryption using scan patterns. *Pattern Recognition 25*, no. 6: 567-581, 1992.

[9] Rahulamathavan, Yogachandran, Raphael C-W. Phan, Jonathon A. Chambers, and David J. Parish. Facial expression recognition in the encrypted domain based on local fisher discriminant analysis. *IEEE Transactions on Affective Computing 4*, no. 1: 83-92, 2013.

[10] Huang, Gary B., Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Vol. 1, no. 2. *Technical Report* 07-49, University of Massachusetts, Amherst, 2007.

[11] Upmanyu, Maneesh, Anoop M. Namboodiri, Kannan Srinathan, and C. V. Jawahar. Efficient privacy preserving video surveillance. In, *IEEE 12th International Conference on Computer Vision*, pp. 1639-1646. IEEE, 2009.

[12] Bertilsson, Michael, Ernest F. Brickell, and Ingemar Ingemarsson. Cryptanalysis of video encryption based on space-filling curves. In *Workshop on the Theory and Application of of Cryptographic Techniques*, pp. 403-411. Springer, Berlin, Heidelberg, 1989.

[13] Xu, Lu, Zhi Li, Jian Li, and Wei Hua. A novel bit-level image encryption algorithm based on chaotic maps. *Optics and Lasers in Engineering* 78 : 17-25, 2016.

[14] Lathey, Ankita, Pradeep K. Atrey, and Nishant Joshi. Homomorphic low pass filtering on encrypted multimedia over cloud. In, *IEEE Seventh International Conference on Semantic Computing (ICSC)*, pp. 310-313. IEEE, 2013.

[15] Suresh, V., and CE Veni Madhavan. Image encryption with space-filling curves. *Defence Science Journal* 62, no. 1: 46-50, 2012.

[16] Zhang, Lan, 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, no. 11: 3258-3271, 2012.

[17] Osadchy, M., Pinkas, B., Jarrous, A. and Moskovich, B., May. Scifi - a system for secure face identification. In, *IEEE Symposium on Security and Privacy (SP)* (pp. 239-254). IEEE, 2010.

[18] Enayatifar, Rasul, Abdul Hanan Abdullah, and Ismail Fauzi Isnin. Chaos-based image encryption using a hybrid genetic algorithm and a DNA sequence. *Optics and Lasers in Engineering* 56: 83-93, 2014.