SECURE REVERSIBLE DATA HIDING IN THE ENCRYPTED DOMAIN FOR PRIVACY PROTECTION IN CLOUD ENVIRONMENT

Submitted in fulfilment of the requirements of the degree of

Doctor of Philosophy

By

Ms. Jeeva K. A

Supervisor: Dr. Sheeba V. S

Department of Electrical Engineering

Government Engineering College Thrissur

UNIVERSITY OF CALICUT

July 2020

UNIVERSITY OF CALICUT

BONAFIDE CERTIFICATE

Certified that this thesis titled **"SECURE REVERSIBLE DATA HIDING IN THE ENCRYPTED DOMAIN FOR PRIVACY PROTECTION IN CLOUD ENVIRONMENT",** is the bonafide work of Ms. Jeeva K.A, who carried out the research under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion of this or any other candidate.

Dr. Sheeba V. S.
(SUPERVISOR)
Professor,
Dept. of Electrical & Electronics Engineering,
G.E.C Thrissur.

Place: Thrissur

Date: 27-07-2020

ACKNOWLEDGEMENTS

Through this little note and limited space, I try my best to express my sincere gratitude to some of those people without whose help this thesis simply would not have come out.

First and foremost, I would like to express my fathomless gratitude to Ms. Sheeba V.S., Principal, G.E.C. Thrissur for allowing me to do my research under her guide-ship, moulding me to a researcher and finally helping me to make the five long years toil become fruitful by materializing this thesis in the present form. Amid her busy routine responsibilities as the head of a large institution, she found time to evaluate, correct and lead my research on the right path by going through each word I put in the paper.

I use this opportunity to thank my Doctoral committee members Ms. K Meenakshy, GEC Thrissur, and Mr. Nithin V George, IIT Gnandhinagar for providing me timely directions in carrying out my research work.

I am much obliged to Mr. M. Nandakumar, Ms.. Reji P and Ms. Preetha K.P who as the Heads of the Department of Electrical and Electronics Engineering helped me by providing the facilities in the EEE department for my carrying out my work.

I am much indebted to Ms. Deepthi P. P, NIT Calicut for introducing such a novel area of research for my work. M/s Vishnu Madhanmohan, Adil Nasser, Remya George and Shwetha C, who have helped me to fulfil this task, are also remembered with many thanks.

Finally, I would like to acknowledge the unconditional love, support and prayers of my family members. Without them, this thesis could not have been accomplished.

Jeeva K. A

ABSTRACT

Revolution in the generation, storage, and communication of digital information has brought profound changes in our society. The digital information age has evolved with numerous opportunities as well as new challenges. The ease with which digital content can be exchanged over the Internet has created copyright infringement issues. Data hiding has emerged as a major research area due to the phenomenal growth in internet and multimedia technologies. It involves concealing confidential data within another seemingly innocuous cover media such as text, video, audio, image, and compression coding. The embedded data can then be used as an authentication code for protecting the intellectual copyrights (Watermarking) or as confidential data for sharing information (Steganography). However, in most of these applications, extraction leads to the destruction of cover media. But there are some sensitive applications like medical diagnosis, military, and law enforcement applications, where even minor distortion in original data may be unacceptable. A slight distortion may lead to a misinterpretation of the final decision. Hence, reversible data hiding schemes are used to restore the cover media without distortion in such sensitive applications.

In the past decade, rapid technological developments in areas such as social networking, online applications, cloud computing and distributed processing in general have raised important concerns regarding the security–and in particular the privacy–of user related content. The privacy problem in the cloud is a severe concern, mainly because the data and signals in a cloud can be distributed among different servers and even different countries with their data protection legislation. This fuzzy nature of processing and location in the clouds can negatively affect the trust that users put on these systems,

as they face the risk of losing control over their data and processes when they are outsourced. Cryptography provides solutions for securing sensitive data by converting it to scrambled noise like signals. But cryptographic primitives like encryption, signature, etc., are designed to protect data at rest or in transmission, but fail if one who processes the data himself is untrusted. For applying common signal processing operations, the data should be taken back to its original form and thus the security and privacy of the data are compromised. So the challenge is to enable the processing of private signals similar to their original form without leaking privacy. Signal processing in the encrypted domain [SPED] allows execution of operations directly on the encrypted signals with no access to them in the plain domain, and problems related with piracy and privacy of digital content can be addressed through the use of technological solutions developed within the emerging field of Signal processing in the encrypted domain.

The thesis presents Paillier encryption based high capacity reversible data hiding algorithms in the encrypted spatial domain and encrypted frequency domain with high perceptibility. The secret data can be directly inserted into and extracted from the encrypted images with the private key which can be used as an authentication code. This is useful for cloud computing services while protecting the privacy of the image in the cloud. The results are compared with state-of-the-art algorithms and found offering higher embedding capacity along with high marked image quality. All the algorithms allow perfect retrieval of original cover media. Performance towards common noise attacks is also evaluated in the encrypted domain. The algorithms have been implemented in C++ with the help of the GNU Multi-Precision library and the NTL library for processing integers of arbitrary length.

CONTENTS

LIST OF ABBREVIATIONS

AES	Advanced encryption standard
AGSP	Accurate gradient selective predictor
bpp	Bits per pixel
CDM	Code division multiplexing
DCT	Discrete cosine transform
DE	Difference expansion
DES	Data encryption standard
DFT	Discrete Fourier transform
DWT	Discrete wavelet transform
e-DCT	Discrete cosine transform in the encrypted domain
FDCT	Fractional DCT
FFT	Fast Fourier transform
LDPC	Low density parity check
LSB	Least significant bit
MIM	Multiplicative inverse method
MSB	Most significant bit
MSE	Mean squared error
PEE	Prediction error expansion
PSNR	Peak signal to noise ratio
PVO	Pixel value ordering
QIM	Quantization index modulation
RDH	Reversible data hiding
RDH-ED	RDH in encrypted domain
RDH-EI	RDH on encrypted images
RLC	Run length coding
RRBE	Reserve room before encryption
SPED	Signal processing in the encrypted domain

SQS	Sequential quantization strategy
SSIM	Structural similarity index metric
SVD	Singular value decomposition
VQ	Vector quantization
VRAE	Vacate room after encryption
WHT	Walsh-Hadamard transform
WQM	Weighted quantization method

TABLE OF NOTATIONS

p	Large prime number
q	Large prime number
N	Product of prime numbers
g	Generator of the Paillier scheme
r	Random number
m	Plain text message
c	Cipher text message
$\mathbf{X}$	Cover image
$E[x]$	Encryption of x
$D[x]$	Decryption of x
b	Binary watermark bit
$\mathbf{A}$	Encrypted cover image
$\mathbf{A}_w$	Encrypted and Watermarked cover image
$\mathbf{X}_w$	Watermarked image in plaintext domain
$\mathbf{F}$	e-DCT of cover image
$\mathbf{F}_w$	Watermarked e-DCT of cover image
$\mathbf{f}$	DCT of cover image
$\mathbf{f}_w$	Watermarked DCT of cover image
$\mathbf{B}$	e- DCT coefficient matrix
$\mathbf{B}_w$	Watermarked e- DCT coefficient matrix
$\mathbf{C}$	Watermarked e-DCT matrix of cover $\mathbf{X}$

$\mathbf{Y}_w$	Watermarked DCT matrix of cover **X**
$\mathbf{Y}$	DCT matrix of cover **X**
$\bar{x}$	Mean of x
σ	Standard deviation

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

We are witnessing the bombarding eruption of 'information technology' worldwide. Information as data signals is flowing through invisible waves enveloping us. From data regarding national security to the simple answer to the quest for knowledge of a common man is processed and delivered in each micro part of a second. A simple handy device that could answer our questions and control our household equipment is not a thing of astonishment anymore. According to the latest statistics [85], 24000 gigabytes of data are uploaded across the world in each second.

The alarming part of this development is that malpractices in this field are also progressing at a proportional rate which could ultimately retard the fruitfulness and growth of technology. New versions of malware, viruses, bots, etc., are introduced by miscreants every day. Despite the tireless efforts of scientists to curb this menace, it is ever growing. Recent studies show that one-third part of total computers in the world is affected by viruses. Anti-virus software and firewalls provide protection up to a certain limit, but still, our valuable data transferred through digital media is under the threat of malicious attack. Social media is believed to be the most popular platform for data communication nowadays and the sickening truth is that it is the most vulnerable place for your privacy. The Federal Trade Commission of the U.S. has recently [86] imposed a fine (still to be approved by Justice Department) to the tune of 5 billion dollars to 'Facebook' for breach of trust to its users. Vital personal details of the users were allegedly handed over to third parties by the social media giant for its culpable business interests. Mobile apps (software application), even though downloaded from reliable app

stores, are said to be the resources for malignant content that could hamper your device or loot your data.

1.1 Motivation

The fatal attraction to the internet due to its flexibility for transferring digital content has paved the way for certain copyright infringement issues. Any copyrighted material can be easily exchanged over the internet without the consent of its rightful owner. Digital watermarking, which is an application of data hiding technique, has been evolved as a feasible solution to protect Digital Management Rights and content authentication and is still a hot topic for researchers.

In classic watermarking techniques, the quality of the cover medium was given the least importance in comparison with the secret data endorsed during the retrieval process. But there are certain scenarios such as medical imagery where every bit of information of cover medium is equally important as the secret code during the retrieval. Reversible data hiding (RDH) addresses this problem and gives equal importance to the cover as well as secret data.

Without mentioning the current status of Cloud Computing, the story of atrocities in the digital communication field would be incomplete. As a resource of abundant storage space and limitless software applications, Cloud computing is spreading its wings over the world, day by day. Any entrepreneur can hire the above-mentioned services offered by cloud storage providers by paying nominal charges and thereby able to reduce their investment in hardware and software procurement required for computational jobs. Apart from the initial investment in the infrastructure, the maintenance cost is also saved since the entire machinery for computation and storage is maintained by the service providers. The proximity of these services makes the cloud storage more appealing to its

customers. From any part of the world, they can access the services and do their job with ease and comfort. As the service provider adopts the latest cutting edge technologies and possesses large scale storage areas for backup and maintenance of data, the possibility for a crash is minimal. Hence, multifaceted advantages such as elasticity, flexibility, economy, and durability make cloud computing unique in the present digital era.

However, this vast utility has its disadvantages. The concerns about privacy and security forbade the wide acceptance of outsourced storage like the cloud. As the bifurcation point of privacy and security is intricate in the cloud point of view, one cannot be compromised for the other. Even so, privacy should have an upper hand in the cloud, because valuable data and signals rely upon different servers that lay across the world, which have different territorial privacy concepts and legislation. Most users find it difficult to depend upon such a fuzzy service, which is prone to lose their control over the data once it is uploaded. According to the legal agreement executed between the client and service provider in the cloud environment, the latter is empowered to access any data entrusted with them. Therefore the user is constrained to believe that service outsourced will be carried out by the provider faithfully protecting his valuable data from any breach of privacy. In other words, the consumer has to assign his tasks to the service provider with blind trust because the privacy of the data and signals placed in a public platform like the cloud can be invaded by service providers themselves or hackers from outside. Once the data is uploaded to the cloud, the customer loses his control over it and anything that happens beyond his premises is invisible to him. It is assumed that the total financial loss met by the business organisations due to breach of trust actions happened in the cloud computing area during the last year sums up to 50 billion dollars.

Hence, it has become evident that privacy issues can constitute a severe barrier to cloud adoption. Classic cryptographic primitives like encryption, signature, etc. are

designed to protect data at rest or in transmission, but fail if one who processes the data himself is untrusted. For applying common signal processing operations, the data should be taken back to its original form and thus the security and privacy of the data being compromised. So the challenge is to enable the processing of private signals similar to their original form without leaking privacy. Signal processing in the encrypted domain [SPED] allows execution of operations directly on the encrypted signals with no access to them in the plain domain, and problems related with piracy and privacy of digital content can be addressed through the use of technological solutions developed within the emergent field of Signal processing in the encrypted domain.

1.2 Research objectives

The objective of this research is to develop privacy preserving secure reversible data hiding algorithms on grey images that can be either used for covert communication or authentication. The algorithms should preserve not only the secrecy of the extra bits added but also the secrecy of the host signal. They should be suitable to be adapted in the cloud environment so that third party outsourcing is possible without leaking privacy. Usually, in a cloud scenario, the client will be outsourcing storage and services from the cloud and authentication of electronic data and its secrecy becomes a big issue.

To meet the above requirements, the following objectives were set.

- To develop data hiding algorithms for authentication suitable in a cloud environment to enable outsourcing to an untrusted third party without leaking privacy.
- To design data hiding algorithms that work on encrypted images to protect the privacy details of the cover that carries sensitive information.
- To enable the embedder to embed data homomorphically on the encrypted cover without knowing the details of the cover used.

- To develop data hiding algorithms with high embedding capacity and high perceptual quality compared to the state-of-the-art algorithms.
- To ensure complete reversibility of the embedding process to enable error-free retrieval of the embedded secret data and cover media.
- To validate the robustness of the designed algorithms towards statistical attacks and common signal processing attacks.

1.3 Research contributions

This thesis presents high capacity image based reversible data hiding schemes which can be evaluated both in the plaintext domain and encrypted domain. We put forth four algorithms to preserve the authenticity as well as the privacy of the multimedia data without fear of an intrusion. All these algorithms use encrypted versions of natural images as a host image on which additional data will be embedded, and allow perfect reversibility on retrieval of these natural images in noise-free conditions. Using these algorithms the content owner could outsource an embedding algorithm from an untrusted third party in a cloud environment without disclosing the cover to the embedder as it contains very sensitive data. The owner encrypts the cover image to maintain the secrecy of the data and embedder embeds the secret data on the encrypted image coefficients homomorphically without knowing the host image. The owner could permit the authorized people with a valid data hiding key for extracting the embedded secret data in the encrypted domain but allow only the intended single recipient with the corresponding decryption key to decode the marked data and reveal the cover. For achieving better robustness against common signal processing operations, algorithms are modified to perform data embedding in the encrypted transform domain. The integer discrete cosine transform in the encrypted domain (e-DCT) is chosen for this. The proposed schemes

give better embedding capacity and image quality when compared with state-of-the-art reversible data hiding algorithms existing in the literature. The algorithms perform well in the presence of some popular noise attacks in the encrypted domain.

1.4 Thesis outline

A brief description of the organisation of the thesis is given below.

In chapter 1, a general introduction and the need for privacy and security in a cloud-based environment is described. The chapter also explains the motivation for the proposed work, outlines the research objectives, and presents the contributions made in this thesis.

Chapter 2 provides the basics of reversible data hiding techniques, the use of encryption for privacy, and an overview of cryptographic schemes. The process of encryption and decryption in two popular encryption schemes RSA and Paillier which exhibit deterministic and probabilistic properties respectively are provided. Finally, a brief overview of signal processing in the encrypted domain and the cryptographic primitives that support the processing of encrypted signals are presented.

Chapter 3 presents a detailed literature review of the state of the art reversible data hiding techniques. It provides a general overview of the conventional RDH techniques available in the literature. The chapter focuses more on the algorithms listed in the encrypted domain. The encryption scheme may be either symmetric or asymmetric and the embedding may be either in spatial or transform domain.

Chapter 4 details the proposed algorithm where embedding is done in the encrypted spatial domain. Correlation analysis to verify the security of the Paillier scheme on different cover images is provided. The ability of the scheme to resist statistical attacks

is also verified. Finally, the performance of the algorithm is analysed and compared with the state- of- the -art algorithms.

Chapter 5 provides the implementation aspects of the discrete cosine transform in the encrypted domain. The embedding and extraction processes of the proposed algorithms in the encrypted DCT domain are detailed here. The results are compared with the encrypted spatial domain embedding techniques as well as encrypted transform domain embedding techniques available in the literature.

In chapter 6, the performance of the proposed algorithms towards some of the possible attacks in the encrypted domain is analysed.

Chapter 7 draws general conclusions derived from the research work and future scope.

CHAPTER 2

REVERSIBLE DATA HIDING-A REVIEW

2.1 Introduction

The impact of enticing developments in digital communication technologies is reflected in every nook and corner of the world with the widespread use of the internet as a communication channel. Being a potential tool for the creation and transmission of information in digital form, the internet provides unfathomable opportunities to the world in contrast to the prevailing communication methods. But, this omnipotent tool has an Achilles heel when it comes to the case of secrecy and privacy. Processing and transmission of multimedia information through the internet suffer from predominant violation of the internet privacy policies which cause a threat to personal privacy. Since visual perception is more effective, the usage of digital images is more proliferated, and its security is imperative. This leads to the evolution of a new domain of technology in digital multimedia processing called data hiding. It involves concealing the confidential data within another seemingly innocuous cover media such as text, video, audio, images, etc. The embedded data can then be used as authentication codes for protecting the intellectual copyrights (Watermarking) or as confidential data for sharing information (Steganography).

In earlier data hiding methods, the process of hiding data in the cover media (embedding) leads to distortion or loss of cover media while trying to retrieve the secret data (extraction) at the recipient's side. However, there are certain sensitive applications such as medical diagnosis, military, and law enforcement applications, etc., where even

minor distortion of the cover media may be unacceptable. For example, a small change in the mammogram image sent to a distant doctor for expert advice may lead to an error in medical diagnosis. Hence, the reversible data hiding (RDH) schemes, also called lossless data hiding schemes, are developed to restore the cover media without distortion for such sensitive applications. In 1997 Barton proposed the first RDH algorithm for authentication of the digital data. Since then several algorithms have been reported in the literature of which many make use of images as their cover medium. The performance of these systems widely varies from one method to another.

The essential properties of RDH algorithms, especially when used for authenticity, are as follows:

- Imperceptibility: The watermark should be perceptually invisible, or its presence should not obstruct the work being protected.
- Robustness: It is the ability to resist intentional attacks by the hacker or unintentional attacks through common signal processing operations such as cutting, cropping, etc.
- Capacity: It indicates the number of secret bits that can be embedded in the host signal.

It can be seen that the above- mentioned parameters are interlinked with one another and improvement in one parameter can be achieved at the cost of compromising the others.

Even in reversible data hiding schemes, until recently, the privacy of the cover image was not given any importance, or rather the criteria was only to keep the secrecy of additional data that is embedded on the cover image. Privacy of the cover image can be protected by encryption which is discussed in the next section

2.2 Encryption for privacy protection

Radical changes in cyberspace have created a virtual world that offers anything at the fingertip of a common man, extending him vistas of knowledge, information, and his daily life requirements, instantaneously. In this virtual world people are not kept aloof by boundaries or walls, instead, tied together with unseen strings of the World Wide Web. They can confer their thoughts with others in another part of the world, without bothering about time and distance. They can procure anything from where it is available on this earth if they have sufficient money to pay off.

It is quite natural to have a negative side for this boon to mankind. As a universal rule, we have to pay for what we get from internet services. To provide the best service, the web browsers have to collect, analyse, and categorise the details of the information sought as well as the person who seeks such information. Not only the search engine providers but also the trackers who follow our searches for their business interests can collect our vital data. Most customers, especially social media users utilise web services to share sensitive private information without any discrimination because they are not aware of these infringements. As a result, their whereabouts, likes and dislikes are extracted by the web browser providers or other unknown third parties. These extorted facts may or may not be used with good intentions by the tracker. It is viable to sell these facts for a ransom. Therefore, a more reliable method is to be adapted for secure data communication through web services.

Cryptography is considered as a viable solution to protect the data for a long time. It changes the data into meaningless random like signal and the authorized receiver can retrieve the signal using a secret key. However, until recently, cryptography was only used as a reliable tool to protect the transmission of confidential data from the sender to a receiver over public channels. At the sender side, the common signal processing

operations such as segmentation, compression, etc. are performed on the original data, it is then encrypted using an encryption algorithm and transmitted. At the receiver side, the received, encrypted signal is first decrypted using the secret key, then the inverse signal processing operations are applied to retrieve the original information. The encrypted data passes through the channel as a stream of random bits so that even if it catches the attention of a hacker, he cannot retrieve the original data without the secret key. Another popular means of securing the data through encryption is storing it in a database. The owner can conveniently use the storage space provided by the service providers in the encrypted form without disclosing the details even to the service provider.

2.3 Basics of encryption

The primary objective of any encryption scheme is to preserve confidentiality. According to Kirchhoff's principle, the security of an encryption scheme must rely on the secrecy of the key, not on the obfuscation of their code. There are two categories of encryption schemes developed based on this principle, called symmetric and asymmetric schemes.

Symmetric key systems use the same key for encryption and decryption. Here the security of encryption lies in the secrecy of the key. Therefore, when using the symmetric key system additional care should be taken to send the common key to the recipient securely. DES, AES, etc. are examples of the symmetric key cryptosystems. The public key cryptosystems, on the other hand, use two different keys for encryption and decryption, and thus avoids the burden of sharing the key securely. However, this advantage is achieved at the cost of complex algorithms and added computational complexity. The well-known schemes like RSA, Paillier, etc. are examples of the public key cryptosystems.

2.3.1 Symmetric/private key encryption

In symmetric encryption schemes, both encryption and decryption are performed with the same key. When two people would like to communicate securely, they would have agreed upon the secret key prior to actual communication takes place. The scheme also demands separate keys for every different pair of users, thus an individual has to remember a large number of keys in his/her daily life. In spite of these difficulties, the symmetric schemes being fast, are advantageous, and find many applications. Block ciphers and stream ciphers are subcategories of the symmetric schemes.

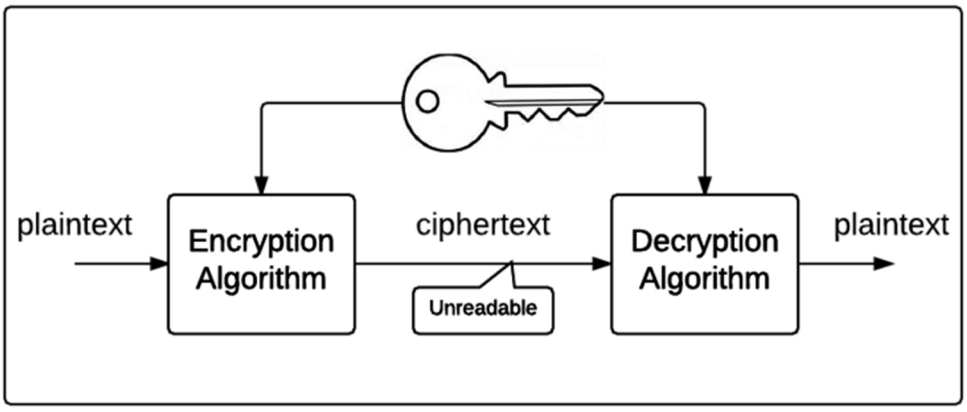


Fig.2.1 Symmetric key cryptosystem

2.3.2Asymmetric/ public key encryption

Asymmetric or public key cryptography is a cryptographic scheme that uses two keys that are mathematically related, for providing security-a public key and a private key. Unlike symmetric key algorithms that use a single key for both encryption and decryption, here each key performs a different unique function. The public key/ encryption key performs encryption and private/ decryption key performs decryption. Though the two keys are related, it is found computationally hard to deduce the private key from the knowledge of the public key. Thus public keys are freely distributed,

allowing the users to encrypt the data and to verify the digital signatures. The private keys are kept secret to ensure that only the owners of the corresponding public keys can decrypt the data and create digital signatures. In other words, the public keys resemble e-mail id in an electronic mail system and private key resembles the corresponding password.

These schemes are more attractive than the symmetric schemes in the sense that it avoids the need for any sort of key exchange between the sender and receiver prior to actual communication. They provide more features than mere encryption. They also have a big drawback that the nontrivial complex mathematical computations make them much slower.

Figure 2.2 shows the working of a public key cryptosystem.

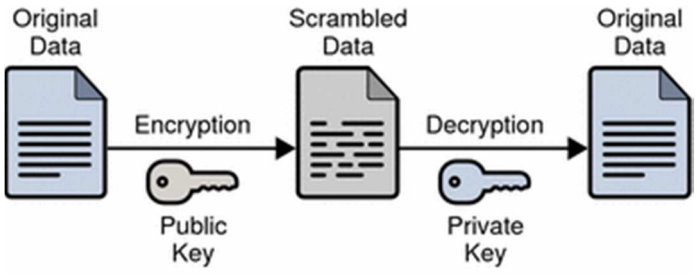


Fig.2.2 Public key cryptosystem

The size of the public key pairs is directly related to the security of the data. Encryptions with higher key sizes are difficult to break and may provide better security. Typical key sizes today used are 1024 or 2048 bits.

Benefits offered by public key cryptography are as follows:

Security/confidentiality: The encryption process scrambles the data into a noise-like signal using the individual's public key and can only be decrypted with the individual's private key. Thus only the intended receiver can decrypt and view the contents.

Authentication: Authentication is achieved through digital signature. Here individual's secret key was used to apply the signature through encryption; the recipient can verify the signature by decrypting with the public key.

Data Integrity: Encryption protects data against attacks and ensures that data is not changed by a hacker through duplication, insertion, deletion, modification, or reordering.

Nonrepudiation: Since the individual is the only one who knows the private key used to apply the signature, he/she cannot deny this identity and claim that it wasn't he/she who put the signature.

The two widely practiced examples of public key cryptosystems are RSA and Paillier which are presented in the next section.

2.3.3 RSA cryptosystem

RSA was invented by Ron Rivest, Adi Shamir, and Len Adlemanin the year 1977 [83]. It is considered as one of the widely accepted cryptosystems for secure communication. It is a public key scheme and uses two separate keys for encryption and decryption. The security of RSA lies in the practical difficulty of factorizing the product of two large prime numbers, which is a hard problem.

As with any similar encryption schemes, the RSA includes three steps: key generation, encryption, and decryption.

The keys for the RSA algorithm are generated in the following way:

1. Select two large prime numbers p and q.
   - For security purposes, p and q are randomly selected numbers and should be of same bit-length.
2. Calculate N as the product of p and q.
3. Calculate $\phi(N) = \phi(p).\phi(q) = (p-1).(q-1)$, where ϕ is the Euler's totient function.
4. Choose an integer e such that $1 < e < \phi(N)$ and e is relatively prime with $\phi(N)$.

5. Find $d = e^{-1} (\mathrm{mod} \phi(N))$; i.e. d is the multiplicative inverse of $e \mod (\phi(N))$.

The pair (e, N) forms the public key that is made available publicly and (d, N) is the corresponding private key that must be kept secret. p, q, and $\phi(N)$ must also be kept secret since the knowledge of these can be used to deduce the secret d.

The encryption and decryption in the RSA algorithm can be performed by the following process [83].

For a message m such that $0 \leq m < N$, the corresponding ciphertext is computed as:

$$c = m^e \, (\mathrm{mod} N)$$

The original message m can be reconstructed from c by applying the private key as:

$$m = c^d \, (\mathrm{mod} N)$$

RSA encryption is a deterministic cryptosystem since the encryption involves no random component. In such cryptosystems, for a chosen encryption key, a chosen-plaintext will always be encrypted as the same ciphertext. Further, transmitting the same message encrypted with the same key multiple times may draw the attention of a hacker easily.

With deterministic cryptosystems, it is not difficult for a hacker to successfully implement a chosen- plaintext attack. He can encrypt probable plaintexts with the public key and see if they produce the same ciphertext. A cryptosystem is considered semantically secure if a hacker cannot differentiate two encryptions from each other even if he has access to the corresponding plaintexts. So, in practice, probabilistic cryptosystems are preferred over deterministic systems for the security point of view. These systems normally involve a random vector called initial value (IV) which changes whenever we encrypt a message.

2.3.4 Paillier cryptosystem

The Paillier encryption scheme is public key algorithm developed by Pascal Paillier in 1999 [84].It is probabilistic in nature and involves a random number in each encryption. The scheme is based on the difficulty in deciding whether a number is an N^{th} residue modulo N^2. The problem is considered computationally infeasible in the cryptography community and is related to the hardness to factorise *N*, if *N*is the product of two large primes.

The scheme includes key generation, encryption, and decryption processes as given below [84]:

Key generation:

Let p and q be two large prime numbers of equal length that are chosen randomly and independently, such that

$$\gcd(pq, (p-1)(q-1)) = 1$$

Let $N = pq$, $\lambda = \text{lcm}((p-1), q-1))$ and g be a random integer ($g \in Z^*_{N^2}$)such that N divides the order of g and $\mu = (\mathrm{L}(g^{\lambda} \bmod N^2))^{-1} \bmod N$ where the function $\mathrm{L}(x)$ is defined as $\mathrm{L}(x) = \dfrac{x-1}{N}$

Then the public or encryption key is (N, g) and private or decryption key is (λ, μ).

Encryption: Let m denotes the message ($m \in Z^*_N$), r be a random number and $r \in Z^*_N$

Now the ciphertext c corresponding to m can be computed as

$$c = g^m r^N \bmod N^2 \tag{2.1}$$

Decryption: From c, the original message *m* can be retrieved as

$$m = L(c^{\lambda} (\text{mod } N^2)).\mu \ (\text{mod} N) \tag{2.2}$$

2.4 Signal processing in the encrypted domain

Signal processing in the encrypted domain (SPED) or Secure Signal Processing is an inspiring new research field drawing the attention of researchers. The need for SPED technologies originates from the growing societal awareness and relevance of security and privacy. With the arrival of social media networks like Facebook, WhatsApp, etc. everyone of us started sharing more and more personal data within larger groups nowadays. The evolution of cloud computing has also gained increasing interest from a commercial point of view. This new concept seems very appealing for service providers as they can earn from hiring out their computation and storage resources. It finds attractive for the users as they can avoid the huge initial investment in the resources by outsourcing their processes and data to a cloud. However, controlling the access and use of these data also has raised big concerns. As already explained in section 2.3, classic cryptographic primitives are designed to protect the data at rest but fail if the processor itself is untrusted. At the same time, the use of personal data becomes more diversified demanding more flexibility in presentation and processing.

SPED allows the processing of sensitive signals at potentially untrusted sites, without or minimally leaking information. In SPED, signal processing operations are performed directly on the encrypted data. The resultant data will be in the encrypted form and an authorized person with a valid secret key can retrieve the actual result.

The principal cryptographic tool to perform encrypted domain computation is a homomorphic cryptosystem. These cryptosystems exhibit the property that the encryption of a function of some variables can be computed by performing either the same or certain other operations on the encrypted version of these variables. A detailed description of homomorphism and other cryptographic primitives are given in subsequent sections.

2.5 Cryptographic primitives for content protection

Apart from the basic encryption process, the other cryptographic primitives which help to process data in the encrypted domain are given below.

2.5.1 Homomorphism

Although the use of encryption algorithms avoid illegal access to the digital data, these algorithms by itself are not sufficient to prevent unauthorized access by an adversary and protect it along its lifetime. Once decrypted, the data lose its security and are susceptible to signal processing attacks. Many public key cryptosystems permit the processing of signals in the encrypted domain. Homomorphism exhibited by these public key cryptosystems enables a data owner to hire a service from an untrusted third party without compromising the privacy and security of the data he owns.

An encryption scheme is said to be homomorphic with respect to an operation * if there exists an operation ($\cdot$) such that for two message inputs, we have

$$D[E[m_1] \cdot E[m_2]] = m_1 * m_2 \bmod N \tag{2.3}$$

Where E[ ] and D[ ] denotes the encryption and decryption operation respectively. Thus a cryptosystem that is exhibiting additive homomorphism maps addition in the plain text domain either to the same or a different operation in the ciphertext domain.

In the Paillier scheme, addition in the plane domain is equivalent to multiplication in the ciphertext domain.

$$\text{i.e. } D[E[m_1].E[m_2]] = m_1 + m_2 \tag{2.4}$$

$$\text{also } D[E[m]^a] = am \tag{2.5}$$

where a is a public integer. Most of the public key cryptosystems exhibit either additive or multiplicative homomorphism. A fully homomorphic system is still a research challenge. Table 2.1 shows popular public key cryptosystems and the type of homomorphism exhibited by them.

Table 2.1.

Homomorphism exhibited by different encryption schemes

Name of cryptosystem	Operation in the encrypted domain	Equivalent operation in the plain domain
Paillier	Multiplication	Addition
RSA	Multiplication	Multiplication
Okamoto-Uchiyama	Multiplication	Addition
Damgard-Jurik	Multiplication	Addition
Goldwasser-Micali	Multiplication	XOR

Paillier scheme exhibits additive homomorphism. Here a multiplication operation in the cipher domain is analogous to addition in the plaintext domain.

Let c_1 and c_2 be two ciphertexts in Paillier scheme for messages m_1 and m_2, then

$$\begin{aligned} D[c_1 \cdot c_2] &= D[g^{m_1} r_1^N . g^{m_2} r_2^N \bmod N^2] \\ &= D\left[g^{m_1+m_2} r^N\right] \bmod N^2 = m_1 + m_2 \end{aligned} \tag{2.6}$$

where $r = r_1 \cdot r_2$.

2.5.2 Blinding

Privacy preserving protocols very often require nonlinear functions that cannot be realised only using homomorphic encryption systems. Many public key systems permit blind computation of the encrypted data.

Blind Computation with Encrypted Data: Let the sender has some data x encrypted with the public key of the service provider and needs to compute the functionality f with the help of him. The sender doesn't trust the service provider and hence doesn't want to reveal the data to him also. If the data is given directly in the encrypted form, the service provider could decrypt it with his private key and get the knowledge of x. So he chooses a suitable random number r and by homomorphic properties computes $E[x + r]$ and sends it to the service provider. The service provider is now able to decrypt and obtain $x+r$ but cannot retrieve x, thus he computes: $E[f(x + r)]$ and sends it back to the sender that contains the required computation. It is necessary that there must exist an f' such that:

$$f'(E[f(x + r)]) = E[f(x)] \tag{2.7}$$

and f' can be applied to the encrypted data.

Fig.2.3 summarizes the sequence of actions for blinding.

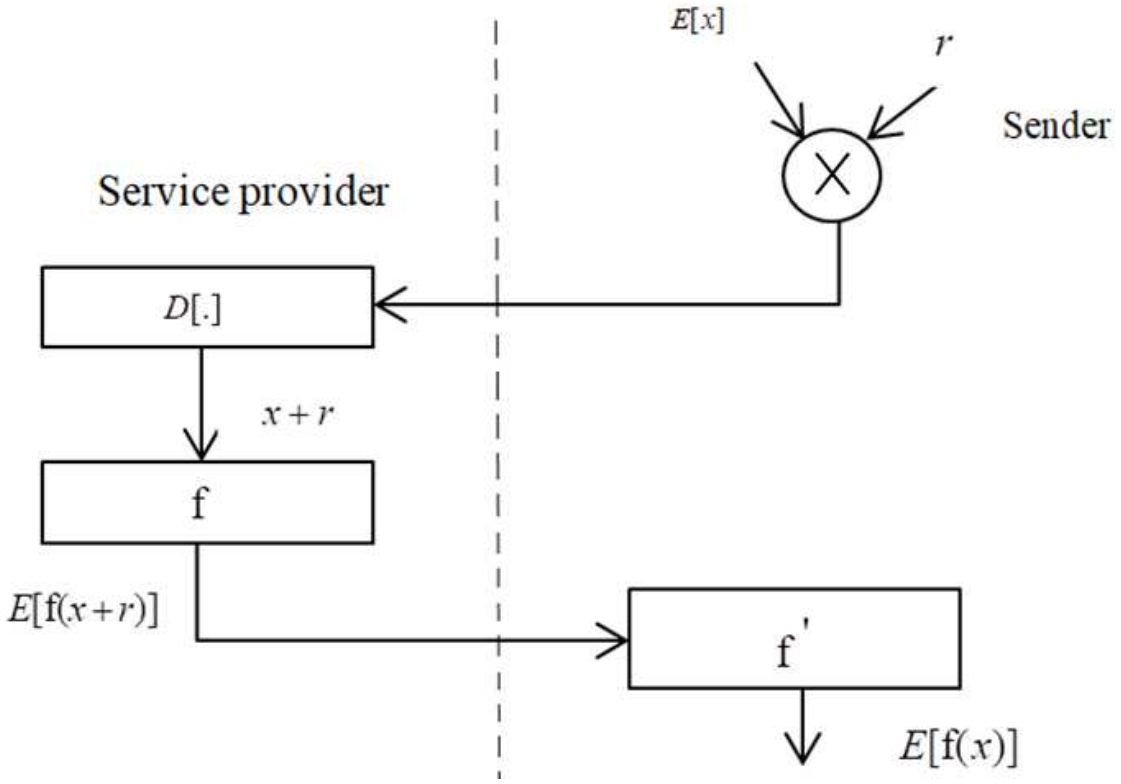


Fig. 2.3 Blind computation on encrypted data

The additive blinding is information theoretically secure, so it provides perfect security on the data, at the same time allowing the computations on the encrypted data. The method is quite often used, and several sub-protocols have been established based on this method.

For example, in modified RSA, blinding involves computing the blinding operation $E[x] = (x.r)^e \bmod N$ where *r* is a random integer between1 and N is relatively prime to *N*. x is the plaintext, e is the public key exponent and N is the modulus. The decryption function $D[c] = c^d \bmod N$ is applied giving $D[E[x]] = (x.r)^{ed} \bmod N = x.r \bmod N$. Finally, unblinding is performed using the function $\mathrm{f}(c) = c.r^{-1} \bmod N$. Multiplying the term ($x.r \bmod N$) by $r^{-1} \bmod N$ reveals the variable x.

The other popular cryptographic primitives offered by secure signal processing are commitment schemes, secure multiparty computation, garbled circuits, etc. Since these topics are out of the scope of this thesis, they are not explained here.

2.6 Applications of secure signal processing

The privacy preserving techniques have been widely used, to realize the systems that can solve a variety of problems. The possible applications of this field are boundless. Some of the interesting scenarios to mention are private database access, in which the client accesses a server through an encrypted query; private data mining, in which two or more parties wish to extract aggregate information from a dataset etc. The data set is formed by the union of their private data and recommender systems, in which users' data is analysed without revealing it.

Secure signal processing finds huge applications in biomedical and signal processing fields. Many signal and image transform are already implemented in the encrypted domain for private data processing without compromising their security. A few of such applications given in the literature are listed below:

- A privacy-preserving system where Bob classifies an Electro Cardio Gram (ECG) signal without learning any information about the ECG signal and Alice is prevented from gaining knowledge about the classification algorithm used by the Server.
- A privacy-enhanced face recognition system which allows us to efficiently hide the biometric using an encrypted version of the widely known Eigen faces algorithm and can keep the result secret from the server that performs the matching operation.
- An ad-hoc system for face recognition in the privacy preserving framework specifically tailored for usage in secure computation.

The privacy preserving computation also finds application in Graph Theory. Graphs are data structures widely used to represent social networks, computer networks, geographic maps, game moves, possible paths in a given environment, and many more. The search for the best first graph algorithm with the help of a heuristic function is also available in the secure domain.

2.7 Summary

The basics of reversible data hiding techniques and an overview of cryptographic schemes are presented in this chapter. The process of encryption and decryption in two popular encryption schemes RSA and Paillier are also discussed. Signal processing techniques in the encrypted domain and the cryptographic primitives that support the processing of encrypted signals are furnished here. For formulating reversible data hiding techniques in the encrypted domain, conventional RDH techniques available in the literature is to be reviewed which is provided in the next chapter.

CHAPTER 3

LITERATURE SURVEY

3.1 Introduction

In the past two decades, the term data hiding has drawn much attention from the research community. Using this technique, we can embed a secret code into a cover medium and the authorized user can extract it later for various applications like watermarking, steganography, etc. The cover medium mentioned here could be either audio, image, or video. In earlier methods, the quality of the cover medium was given the least importance in comparison with the secret data endorsed during the retrieval process. But there are certain scenarios where the quality of cover medium is equally important as the secret code retrieved. Reversible data hiding, which is emerged as a solution to this problem, tries to recover both the cover medium and secret data losslessly at the receiver side.

3.2 Reversible data hiding techniques

In 1997 Barton proposed the first RDH algorithm for authentication of the digital medium [1]. Since then several algorithms have been reported in the literature which make use of images as their cover medium [2-4].

Depending on the domain in which secret data is embedded, the RDH algorithms can be implemented either in the spatial domain or frequency domain. The algorithms in these domains can be further classified as 1) compression based 2) histogram based 3) quantization based 4) expansion based and 5) encryption based.

Compression based reversible data hiding: Compared to the conventional data hiding techniques, the RDH techniques need more space for embedding because along with the

secret code, additional data for reconstructing the cover medium should also be included. In compression-based methods, this additional space required is created through compressing a portion of the cover image for inserting the additional data. Many schemes in literature compress the data directly in the spatial domain or frequency domain for creating extra space for embedding. LSB based data hiding and its variants are very popular in spatial domain schemes [5-8]. Frequency domain based techniques make use of discrete cosine transform (DCT), discrete wavelet transform (DWT), etc. for the same purposes [9-17].

Histogram based reversible data hiding: Extensive research has been already carried out in the field of histogram based reversible data hiding [18-24]. These techniques generally make use of image histogram modification for embedding the secret data. The algorithm may use the histogram of the entire image or block based histograms for modification. Difference histograms based embedding and modulo 256 based embedding have particularly caught the attention of the research community in this category. Some other techniques utilize spatial correlation in natural images for embedding the additional data [25-26]. These methods keep one sub-sampled version of the cover image as reference and modify the histogram of other sub-sampled versions of the same cover image to hide the data. Finally, the reference image is used to retrieve the cover and secret data from the embedded version.

Quantization based reversible data hiding: Data hiding using conventional Quantization index modulation (QIM) is not suitable for RDH due to the unavoidable and irreversible distortion that is associated with the quantization process. However, a variant of QIM called nested QIM with fractional DCT (FDCT) ensures the retrieval of the cover image for bio-medical applications. Sequential quantization strategy (SQS), weighted

quantization method (WQM) and vector quantization (VQ) are other useful techniques based on quantization for reversibly embedding data [27-31].

Expansion based reversible data hiding: Tian in 2003, presented a high capacity RDH scheme based on difference expansion (DE) which is considered as a breakthrough in the reversible data hiding techniques [32]. He could achieve an embedding capacity of 0.5 bit per pixel (bpp) at relatively low computational cost along with low distortion. DE technique has been widely investigated later and improved by many researchers. The improvements were mainly focused on the characteristics of integer transformation, prediction error expansion (PEE), and adaptive embedding and their combinations [33-40].

Encryption based reversible data hiding: Encryption plays a crucial role in RDH when privacy becomes a major concern. In encryption based reversible data hiding techniques, the cover image is normally in the encrypted form. The encryption scheme may be either symmetric (private) or asymmetric (public). The embedding operation may be performed either in the encrypted spatial domain or encrypted transform domain. The receiver extracts the secret data either in the encrypted domain or plain-text domain using the data hiding key. Conversion to plaintext domain is performed through decryption. The original cover is also extracted in the plain domain. A detailed discussion about this topic is given in the following section [41-75].

3.3 RDH in the encrypted domain

Until recently, cryptography was only used as a reliable tool to protect the secure transmission of confidential data from the sender to a receiver over public channels. Many of the cryptosystems are found to be deterministic. The term deterministic means that, for a fixed encryption key, a particular plaintext data will always map to a fixed

ciphertext. These direct mapping will leak partial information to the adversary and make tampering comparatively easier irrespective of the key size. To avoid this issue, we prefer the encryption schemes to be probabilistic which support the possibility of multiple ciphertexts for a single plaintext. This requires that the ciphertexts should be greater in number compared to the number of plaintexts and ciphertexts should not be shorter than plaintexts. Recently probabilistic public key systems find increasing applications in secure data hiding irrespective of their data expansion since they provide semantic security and eliminate the need for key distribution which is a major concern of the symmetric system.

Various combinations of data hiding and encryption schemes for embedding secret data in images for privacy preserving applications are discussed in the literature. In general, the various schemes in RDH in the encrypted domain(RDH-ED) already proposed earlier either reserve room before encryption(RRBE) for embedding secret data [41-51] or vacate the room after encryption(VRAE) for the same purpose [52-60].

Schemes that reserve room before embedding pre-process the cover image before encryption to create space for embedding. Thus RDH with RRBE implementation demands extra work from the content owners, and makes it unsuitable for cloud based storage applications.

Literature provides different techniques for achieving extra space in embedding in the RRBE method. Manipulating the least significant bits (LSB)of the image is one of such techniques. In [41], the suggested method empty room by embedding LSBs of some pixels into other pixels and then use these positions in the encrypted image to carry the extra information. In [42], the authors exploit the alpha channel of the PNG image to embed the secret data. The algorithm divides the secret data into segments. From each segment, one bit is embedded into the LSB of the encrypted pixel, and the remaining bits

are hidden in the corresponding element of the alpha channel. Encryption is performed through two chaotic maps. The algorithm in [43] uses stream cipher based block encryption. It is then pre-processed with run-length-encoding (RLC) and Huffman coding to separate into two sets based on their encoded lengths. To release extra space for embedding the secret data, blocks with larger encoding lengths undergo LSB compression and for the blocks with shorter encoded lengths, their pixel differences are compressed by RLC.

Another method that creates vacant space in the encrypted images for data hiding is prediction techniques [44-47]. Literature gives lots of variants of this method for embedding secret data in the encrypted images. These algorithms generally require a large amount of auxiliary data to restore the original image. In [44], two algorithms called joint method and separable method are suggested by authors. In the joint method, data extraction and image recovery are highly connected. The algorithm allows data extraction only if the receiver has both the data hiding and decryption key. Encryption uses a standard stream cipher with an exclusive–OR operation. This algorithm uses the correlation of a pixel with its four neighbouring locations in predicting the qualified sets of locations. In the second method, data extraction and image recovery are separable. The embedding algorithm is the same as that of the joint method. It permits data extraction from the encrypted image with the help of data hiding key and marked image recovery with the private key.

The algorithm suggested in [45] embeds the data in the encrypted image by estimation technique. A major portion of the pixels is used to approximate the rest before encryption. The encryption is performed with the AES algorithm. The estimated errors are then encrypted with a special encryption scheme. The data extraction and image recovery are free of errors for all the images. The algorithm in [46] uses adaptive bit-

level data embedding method to embed secret data into the encrypted image. It also uses an adaptive check board based prediction method to predict upto 75% of the pixels based on the remaining 25%. The full reversibility of image is possible for this embedding capacity.

In [47], the authors demonstrate an RDH algorithm with pixel prediction and additive homomorphism. They use an accurate gradient selective predictor (AGSP) to determine the original pixel predictions to construct pre-processed pixels. The predictor operates on the nine neighbouring pixels of the chosen pixel, to estimate the gradients in four directions. RC4 is the encryption algorithm used and two location maps are generated, one during encryption and another during data hiding and are embedded into the cover image. Arithmetic coding is used to compress the location maps losslessly. The algorithm needs supplementary information to be embedded in the image like the number of secret digits, capacity parameter, the threshold, the sizes of two compressed location maps, etc. to recover the embedded secret data and cover image which reduces the perceptual quality of the marked image.

The conventional difference expansion scheme proposed by Tian has been modified for encrypted images in [48]. In this scheme, each pixel is separated into two parts: an even integer and a bit, so that the sum is equal to the pixel value. They are encrypted using the Paillier scheme. The encrypted values of the second parts of two neighbouring pixels are changed to conceal an additional bit. Though the scheme claims good embedding capacity, it introduces overflow in the plain domain and hence not suitable for image-based data hiding.

A modified scheme using difference expansion is given in [49].The algorithm uses a pair of pixels in the original image and calculates their average value say, ℓ. With the help of the difference data in samples taken as d, the pair of values is modified as $\ell + d$

and ℓ- d. These pairs are encrypted using the Paillier scheme and send to the embedder. The embedder modifies the first data in the pair if the additional data is a binary '1' and leave it as such if the additional data is a'0'. The pair of values on decryption gives a combination (odd, even) for secret bit 1and (even, even) or (odd, odd) for the pair of values if the additional bit is 0. The authors made a good attempt to avoid the overflow problem existing with the method suggested in [48] but their algorithm fails to provide a good payload. Even with the considerable extra effort required for pre-processing the cover and irrespective of the nature of the cover image chosen, the algorithm inserts only one extra bit in the pair of pixels. The upper limit of the embedding capacity is bounded by 0.5. The algorithm in [50] uses a reversible embedding based on modulo 256 additions. The scheme preserves the mean value of a column of elements before encryption to embed one secret bit in a column of 256 pixels.

Some hybrid schemes are also found in the literature that uses combinations of the techniques for embedding data in the encrypted domain. In [51],the authors present a patch-level sparse representation based method for data embedding. The given cover image is divided into patches and a dictionary based representation is performed using sparse coding. The smoother patches with lower residual errors provide the room for embedding. The sparse coefficients of these selected patches and the corresponding residual errors are encoded and embedded into the remaining patches. The algorithm uses K-means singular value decomposition method to create the dictionary through training procedure. With all these complex procedures, the algorithm provides an erroneous recovery of the embedded data.

The schemes that reserve room after embedding are found more popular among the researchers as they avoid the need for pre-processing. The various algorithms listed in this category generally differ in the domain in which data extraction is performed. The

encryption algorithms used various methods for randomizing the input. In [52], authors use stream cipher encryption which utilizes X-OR operation with pseudo-random bits. The data-hider segments the encrypted image into several non-overlapping blocks. The total bits in these blocks are divided into two sets according to the data hiding key. Embedding is performed by flipping the least 3 significant bits in the first set or second set depending on whether the secret bit is a'1'or '0' respectively. The algorithm in [53] is an improvement brought on [52]. The algorithm makes use of block smoothness for improved data extraction and data recovery. The algorithm also uses the side-match scheme to further reduce the error in data recovery. A further improvement of the above two suggested algorithms, is presented in [54]. In this paper, the authors propose a method for evaluating the complexity of the image blocks considering multiple neighbouring pixels according to the locations of different pixels. The algorithm could reduce the bit error rate when compared to [52] and [53] but could not eliminate it completely. The algorithm in [55] suggests the AES scheme used in CTR mode for encryption process and embedding is performed through bitwise X-OR operation with the sub-blocks of the encrypted image. At the decoder side, the algorithm uses a two-class SVM classifier to discriminate encrypted and non-encrypted image patches. An error correcting code such as Hamming code is also used to reduce the bit error during the recovery process. The algorithm claims a high embedding capacity for sub-blocks of smaller sizes at the cost of an increased error rate.

The algorithms listed in [52-55] extract the hidden data only in the plaintext domain and uses the spatial correlation between the adjacent pixels or blocks. In these methods, the embedded cover image is revealed before hidden data extraction.

In the schemes where extraction is performed in the cipher domain, the hidden data can be extracted with the valid data hiding key without exposing the approximated

cover used. Thus an authentic receiver who possesses data hiding key and the decryption key can retrieve the secret data in the cipher domain and reveal the cover through decryption. The extra space for embedding secret data was created through various methods like local histogram shifting [56], bit compression techniques [57-58], etc. The scheme proposed in [56] performs multi-granularity encryption on the cover image to make it meaningless. Permutation of blocks in an image is done through the coarse-grained encryption whereas fine-grained encryption permutes the pixels within a block. Conventional histogram shifting technique is applied in each block by choosing two randomly selected pixels in each block which represents peak points. The algorithm supports error-free recovery of hidden data but provides low embedding capacity. In [57], the cover image is first encrypted with a stream cipher. The algorithm compresses a part of the encrypted image with LDPC coding to find space for embedding extra data. The compressed data along with additional data are inserted back to the encrypted image. A receiver with the data-hiding key can extract the modified data. By exploiting the compressed data and the additional information provided by the unchanged data, the receiver can construct the plaintext image successfully. In [58], chaotic encryption is performed in the encryption phase and the least significant bits of pixels in the encrypted image are losslessly compressed to reserve space for secret data. Lossless compression is achieved through the Hamming distance calculation between the LSB stream and auxiliary stream. The above-reported algorithms support error free recovery but with a low embedding capacity.

In [59], the authors suggest two different RDH methods through the correction of prediction errors and with embedded prediction errors. They propose a most significant bit(MSB) substitution in contrast to popular LSB substitution and claim that predicting MSB is easier than LSB in the encrypted domain. As the values of the MSB bits are

modified during the data hiding phase, accurate prediction of these values is necessary during the decoding phase. The first approach suggested by authors (called CPE-HCRDH) is not fully reversible but offer an embedding capacity of 1 bpp. In this method, the authors first analyse the original image content, detect all possible prediction errors based on previously decoded neighbourhood pixels, and create an error location map. The image is then encrypted with a Piecewise Linear Chaotic Map and pseudo-random sequence generator. For embedding, the pixels of the encrypted image are scanned first horizontally and then vertically. The MSB of each selected pixel is replaced by one bit. During the retrieval phase, the pixels of the marked encrypted image are scanned in the same order. For each pixel, X-ORing the marked encrypted value with the associated pseudo-random sequence reveals the seven LSB bits. An error location map is used to predict the remaining MSB value. The second approach (called EPE-HCRHDH) concentrates on the exact recovery of the original image sacrificing the payload to a very low value (<0.1). Here the error location information is embedded in the encrypted image and only the remaining pixels are used for embedding the secret. The original image can be reconstructed with the help of the location error information. Though coming under VARE schemes, both the methods demand to pre-process the original image to create an error location map that is used for MSB prediction. Further, these methods make a severe compromise on retrieved image quality for a high embedding capacity and vice versa.

A pixel value ordering (PVO) based RDH scheme is explained in [60] which uses RC4 based homomorphic encryption. The principle behind PVO-based RDH is the prediction error histogram shift. The method also uses a location map for facilitating the reversibility of the original image. Homomorphic encryption used in this scheme does not create further data expansion. The size of the location map depends on the cover image

and increases for images with a high spatial activity (e.g., Lake), and much of the storage space will be consumed by the location map itself leading to a low embedding capacity.

The above mentioned VRAE schemes generally offer low embedding rates since the encryption process leaves very little redundant space in the original image for inserting secret data. In contrast to this, RRBE schemes allocate space for secret data in the plain image itself and transfer this space to the encrypted domain. They offer a better embedding rate compared to VRAE schemes, but the data owner has to bear the additional burden of pre-processing the cover image for reserving space which is impractical without the knowledge of the exact embedding process.

There are some off-the-road experimental research papers seen in the literature for reversible data hiding. An RDH scheme that uses the Chinese Remainder Theorem to prove the ownership in a cloud is given in [61]. The algorithm uses a secret sharing scheme that divides the original data into multiple encrypted shares and embeds the secret information into these shares. Since pixels in the original image are highly correlated, to obtain obfuscated shares of the cover media, randomization and scaling are performed on these pixels so that the neighbouring pixels get uncorrelated. Further, the affine transform is applied to obtain more randomised, noise-like shares that reveal no information. The owner specific secret data embedding is performed on the selected region of interest using the singular value decomposition(SVD) method. The scheme provides approximately constant image quality for various embedding rates on the selected region of interest but does not guarantee a constant embedding strength on different regions of interest chosen. The same authors suggest another RDH algorithm in [62] using Shamir's secret sharing scheme. The scheme makes the information secure by distributing it into multiple random-like secure shares. These shares are obtained based on Shamir's secret sharing scheme and embedding is performed homomorphically based on a secret key. The

selected shares for embedding are decomposed using discrete wavelet transform upto n levels using a different secret key. A reference image is constructed using one of the sub-bands and retaining only the significant coefficient values. Embedding is performed on the reference image using SVD. This new scheme supports a high embedding capacity (upto 2 bpp) maintaining a uniform peak signal to noise ratio (PSNR) which is below 40 dB.

The aforementioned RDH methods on encrypted images (RDH-EI) use various symmetric cryptosystems for encryption. In fact, Public key based RDH-EI systems, with probabilistic and homomorphic properties, are more suitable for privacy-preserving applications. They allow us to conduct operations directly on ciphertexts without any information leakage and key sharing and are ideal for cloud-based applications where data privacy is a major concern.

Several studies are recently reported in the literature on RDH-EI using the Paillier cryptosystem [65-70]. In [66], authors embed secret data on a pair of pixels of the encrypted image by shifting the histogram of absolute differences. Considering two pixels as a group for encryption, the data hider can retrieve the absolute differences of groups of two pixels with the help of a modular multiplicative inverse method. The scheme avoids pre-processing operations on the original image. Additional data can be embedded directly into an encrypted image by shifting the histogram of the absolute differences. The pair of pixels is selected either on an adjacent basis or on a random basis. At the receiver side, the genuine receiver can extract the hidden data from the marked encrypted histogram through the inverse histogram process and the original image can be recreated with the help of marked absolute differences obtained after decryption. The algorithm in [67] requires preprocessing of the plaintext image to reserve room before encryption through LSB substitution. The preprocessed image is then encrypted, divided into groups,

and mirroring is applied to embed the additional data. The retrieval process utilizes the multiplicative inverse method to separate hidden data with the help of prediction error and a mapping table. In [68], X. Zhang *et*.al. propose three schemes-a lossless, a reversible, and a combined scheme using multilayer wet paper coding to embed bits into bit-planes of the ciphertexts. In the first method, ciphertext values aremodified for embedding the additional data into the LSB-planes of ciphertext pixels. Using this method, the data extraction can be done only in the ciphertext domain and it introduces distortion in the extracted original image. In the reversible scheme, histogram shrink is implemented before encryption, and one-half of the ciphertext pixel values are modified for data embedding. The retrieval process introduces errors in the extracted hidden data. The combined method is reversible and allows data extraction partially in the encrypted domain and remaining in the plain domain. The common drawback of the above algorithms is that they provide a low embedding rate which is a crucial performance measure in RDH systems. Some algorithms are reported in the literature that claims a high embedding rate of 1 bit per pixel (bpp). In [69], data hiding is achieved through histogram expansion and shifting in the cipher domain. The histogram of the original image is expanded to create space for data embedding. The secret data can be embedded into the emptied bins. The authors propose two different extraction methods, lossy and lossless, and the size of the secret data that could be embedded relies on the extraction method. Only the lossy method achieves an embedding rate of 1 bpp at the expense of distortion to the recovered host image. In the reversible data hiding technique reported in [70], two algorithms in the encrypted domain are proposed. Here, the first algorithm alters the encrypted pixel values of the image to hide the secret data and the second algorithm uses self-blinding to modify the selected pixel according to the data to be hidden.

However, the methods impose restriction in the domain in which the data retrieval is possible.

The algorithm in [71] illustrates a full embedding scheme on public key encrypted images. Here each image pixel value is represented as the sum of two separate grey values and is separately encrypted. For each pixel, corresponding encrypted pairs are arranged in descending order for embedding a secret bit 1. For embedding bit '0', the values are arranged in the reverse order. The process is repeated for all the pixels in the image and concatenated to get the marked image. At the receiver side, the secret data can be extracted by a simple comparison of the adjacent pairs of values. Though this method is a simple technique offering an embedding capacity 1 bpp, the embedding process doubles the size of the encrypted image. In [72] authors present the conventional difference expansion based RDH technique to the encrypted domain using homomorphic encryption. Here each pair of pixels will be able to carry one bit of secret data giving an embedding capacity of maximum 0.5 bpp and the algorithm also causes data expansion. The algorithm in [73] applies DE based RDH with homomorphic embedding. To reduce the computational complexity, authors use the Chinese remainder theorem (CRT).The algorithm supports an embedding capacity of 1 bpp and offers a perceptible quality of 52 dB or less.

Xianyi *et*. al. in [74] propose encrypted signals-based reversible data hiding using the code division multiplexing (CDM) and value expansion. With only the CDM method used to embed secret bits, embedding rate upto 3bpp can be achieved, but beyond 1 bpp, a lossless retrieval of cover image is not possible. When using CDM along with the value expansion method, the embedding capacity can be further increased with increased distortion of the retrieved marked image. The major drawback associated with the method

is the huge expansion of the data, a minimum of six times, to achieve this high embedding rate.

The algorithm proposed in [75] claims a full embedding capacity of 1 bpp on standard gray scale test images with lossless recovery. It employs the concept of signal energy transfer. Here a signal is represented as the sum of other signals. The signal energy is transferred to these elements and each unit is separately encrypted using the Paillier scheme and concatenated to construct the encrypted image. The embedded data is concealed into these elements using the homomorphic property of the encryption scheme. The reverse operations are performed at the receiver side to recover the image and data. The proposed method also introduces a multiple of 6 times data expansion to the original image which is not a desirable factor both in the storage as well as the communication point of view.

So far all the methods described used algorithms that operate in the encrypted spatial domain. All these algorithms manipulated the encrypted pixel values of cover images to conceal the secret data. Watermarks implemented in the encrypted spatial domain are susceptible to common geometric attacks. Recently watermark embedding in the encrypted transform domain is increasingly drawing the attention of the research community working on the secure signal processing field. Encrypted transform domain implementations inherit all the advantages of their spatial transform domain counterparts.

The next section describes the state-of-the-art of RDH systems in the encrypted transform domain.

3.4 RDH-ED in the transform domain

The implementation of popular image transforms in the encrypted domain is reported in the literature by many researchers. Bianchi et al. propose the implementation of discrete

Fourier transform (DFT) and fast Fourier transform (FFT) in the encrypted domain [76-77]. They also computed popular discrete cosine transform (DCT) in the encrypted domain [78]. Zheng et al. propose the implementation of discrete wavelet transform (DWT) in the encrypted domain and suggest a solution to the data expansion problem through multiplicative inverse method (MIM) [79].

Though watermarking techniques implemented in the transform domain are more robust compared to spatial domain implementations, very few studies have been reported that implement watermarks in the encrypted transform domain. Generally, the watermark algorithms implemented in the transform domain offer a higher visual quality of the watermarked image than the others. In [80], along with the computational details of implementing Wavelet transform in the encrypted domain, authors demonstrate image watermarking using Haar wavelet in the encrypted domain. After performing a five-level wavelet transform in the encrypted domain, Gaussian noise which is chosen as the watermark is added into the low frequency coefficients. In order to recover the desired watermarked image, MIM is applied to the encrypted version of the watermarked image. A direct decryption process without MIM would result in noise like image due to decryption error induced by large data size. In this paper, the authors do not provide a quantitative analysis of the image quality variation with respect to the size of the watermark. Implementation of the Walsh-Hadamard transform (WHT) in an encrypted domain and a watermarking method based on the same transform was suggested in [81]. The additive homomorphic properties of the Paillier scheme are exploited to implement WHT in the encrypted domain. Authors suggest that WHT is especially suitable to implement in the encrypted domain as its transform matrix consists of only integers and hence will not introduce quantization errors. Watermark embedding is performed by modifying the relations among neighbouring transform coefficients. For this, the image

is encrypted and segmented into blocks and WHT is applied to each block in the encrypted domain. A cardinal point is identified in each block and surrounding transform coefficients are modified to embed one watermark bit. Finally, inverse WHT is applied to each modified block and combined to get the watermarked image. The algorithm permits blind extraction based on the correlation existing among the modified coefficients. The authors embedded one bit secret data in a block of size 8x8 receiving a visual quality of 43.18 dB in the reconstructed marked image. The method permits secret data extraction both in the ciphertext domain and plaintext domain. In [82], Guo *et*. al propose a hybrid transform domain based secure watermarking scheme for images. The method utilises the multi-resolution characteristics of DWT and the energy-compaction characteristics of DCT to embed the watermark. The encrypted domain DWT is applied on the original image to obtain four coefficient sub-bands LL; HL; LH and HH, in the encrypted domain. Embedding is performed on the LL band. The LL sub-band is further split into non-overlapping m x m blocks and each coefficient block is splitting into two pairs of similar sub-blocks as given in Fig. 3.1.DCT is then performed on each sub-block and watermark is applied on mid-band DCT coefficients. To embed a watermark bit, the coefficient in one sub-block is modified keeping the coefficient in the paired block as a reference. The proposed algorithm could embed 4 watermark bits in every8x8 block giving an embedding capacity of 0.0625 bpp.

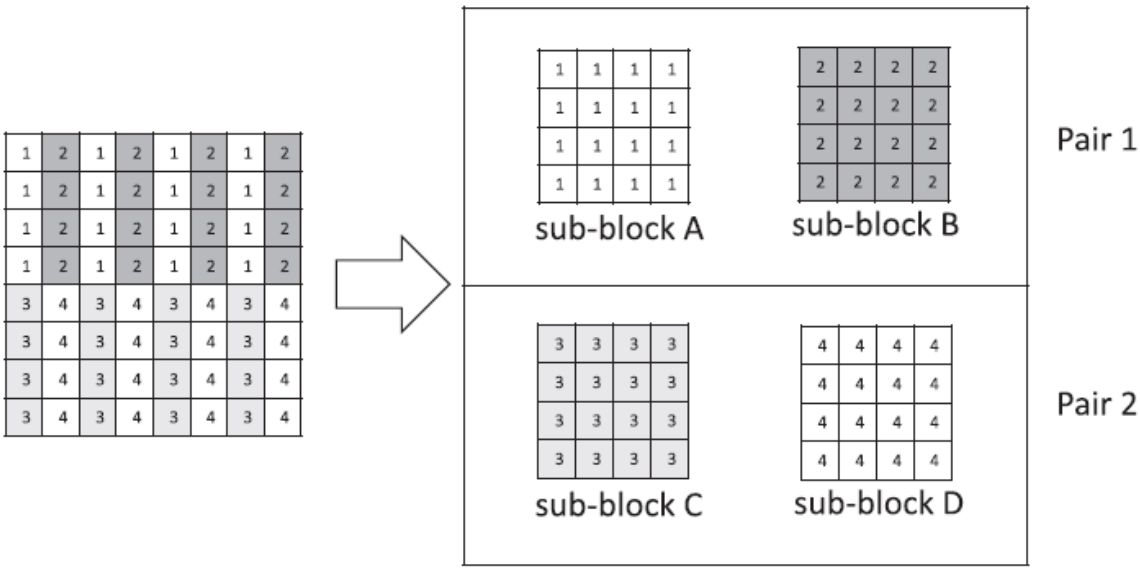


Fig. 3.1 Coefficient block splitting in Guo *et*.al's algorithm

Algorithms in [80-81] use the Paillier scheme to encrypt the image and allow an untrusted third party to embed a watermark in the transform domain without tampering the original images. Blind watermark extraction is possible in the plaintext domain and cipher domain in these methods. However, these algorithms exploit the relations among adjacent transform coefficients for their implementations. An obvious disadvantage of such correlation based watermarking techniques is their fragility to resist statistical attacks [82]. Further, correlation-based embedding methods provide low embedding rates as these algorithms use unmodified reference pixels to facilitate data retrieval.

The proposed algorithms detailed in the subsequent chapters implement RDH on encrypted cover images. Secret data embedding is done either modifying the encrypted pixel values directly(encrypted spatial domain embedding) or modifying transform coefficients computed in the encrypted domain(encrypted transform domain embedding).Since the original cover need not be exposed to the embedding process, the algorithms permit outsourcing the embedding process still preserving the privacy and are ideal for the cloud environment.

3.5 Summary

Different RDH techniques in the encrypted domain are reviewed in this chapter. The embedding capacity is low in all these cases. In this thesis, algorithms are developed to implement RDH in encrypted cover images both in the spatial domain and transform domain which is discussed in subsequent chapters. Since the original cover need not be exposed for the embedding process, the embedding process using these algorithms can be outsourced still preserving the privacy and the method is ideal for the cloud environment.

CHAPTER 4
RDH IN THE ENCRYPTED SPATIAL DOMAIN

In recent years, digital image watermarking has emerged as a potential tool against copyright violations in multimedia contents. In conventional watermarking schemes, the watermark embedder must be the owner of the cover medium if the cover carries private data related to the owner. Signal processing in the encrypted domain also referred to as secure signal processing allows the processing of the encrypted data by special signal processing operations and thus avoids the disclosure of confidential information when the owner depends on a third party for embedding operation.

4.1 Introduction

In this chapter, a robust image-based reversible watermarking in the encrypted domain is presented. For all algorithms presented in this thesis, the Paillier encryption scheme is chosen for encrypting the cover with a key size of 1024 bits to ensure practical security. The algorithm offers a high embedding rate and embeds the data in a homomorphic encrypted domain which eliminates the need for exposing the cover details to a service provider. Watermark bits that carry owner information, are embedded in the encrypted pixel values of the cover image directly. The algorithm exploits the self- blinding property of the Paillier scheme to accomplish flexibility in the extraction domain. Using these algorithms, blind and error-free watermark extraction is possible in the plaintext domain and encrypted domain. The data hiding key can be used to retrieve the hidden data in the encrypted domain itself. The genuine receiver can retrieve the marked image, host image, and the secret data in the plain domain using both decryption and data hiding keys. The algorithm finds applications in privacy preserving distributed signal processing which is a major requirement in the cloud environment.

4.2 Performance measures of encryption schemes

The original cover image is encrypted using the Paillier scheme to protect privacy. The amount of randomness in the encrypted data is a measure of the security of the cover image. Qualitative evaluation of the security of encryption can be analyzed by observing the visual perceptibility of the encrypted image. Figure 4 shows that the encrypted images are incomprehensible and does not reveal any details regarding the content of the original image. Quantitative evaluation of the security of encryption is carried out using histogram analysis, Peak Signal to Noise Ratio (PSNR), and correlation coefficient.

The PSNR of the encrypted image with that of the original is only 8.6338 dB in Lena and is 8.4789 dB for Baboon which ensures that reconstructing the original image from its encrypted version is almost impossible without the decryption key.

Randomness in the encrypted image makes the statistical attacks difficult for cryptanalysts. In a statistical attack, a cryptanalyst explores the existence of predictable relationships between the original data and its encrypted form. To prove the existence of randomness in the input data, the correlation coefficient is used here, which is computed as follows.

$$R_{x.y} = \frac{1}{n}\sum_{i=0}^{n}\left(\frac{x_i - \bar{x}}{\sigma_x}\right)\left(\frac{y_i - \bar{y}}{\sigma_y}\right)$$

where n denotes the total number of tuples (x_i, y_i) in the input data in a given direction, $\bar{x}$ and $\bar{y}$ represent the mean, σ_x and σ_y represent the standard deviation of x_i and y_i respectively. Table 4.1gives the correlation existing in the original image and encrypted image in horizontal and vertical directions. The low value of the correlation coefficient in the encrypted image guarantees no leakage of information from the encrypted image.

Table 4.1

Horizontal and vertical correlation in the original and encrypted image

Cover image	$R_{x,y}$	Original image	Encrypted image
Lena	Horizontal	0.9505	-0.0035
	Vertical	0.8937	-0.0040
Baboon	Horizontal	0.8810	0.0011
	Vertical	0.8835	0.0014

The scatter plots of the distribution of neighboring pixels in the original and encrypted image 'Lena' are given in Fig. 4.1 and 4.2. It shows that pixels in the encrypted image are highly uncorrelated when compared to that in the original image. Fig.4.3 shows the histograms of the original image and the encrypted image. The same analysis is repeated for image 'Baboon' and the results are presented in Fig. 4.4 to Fig. 4.6. Histogram of the encrypted images provides nearly uniform distribution and makes histogram based statistical attacks unsuccessful. All the above demonstrations ascertain that little or no predictable relationship existing between the images and the encryption algorithm can withstand statistical attacks.

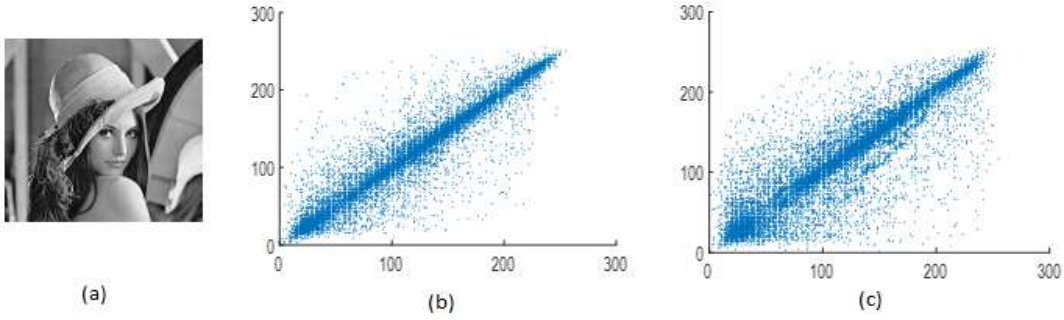


Fig.4.1 Correlation in plaintext image 'Lena' in different directions (a) Image Lena (b) Horizontal correlation (c) Vertical correlation

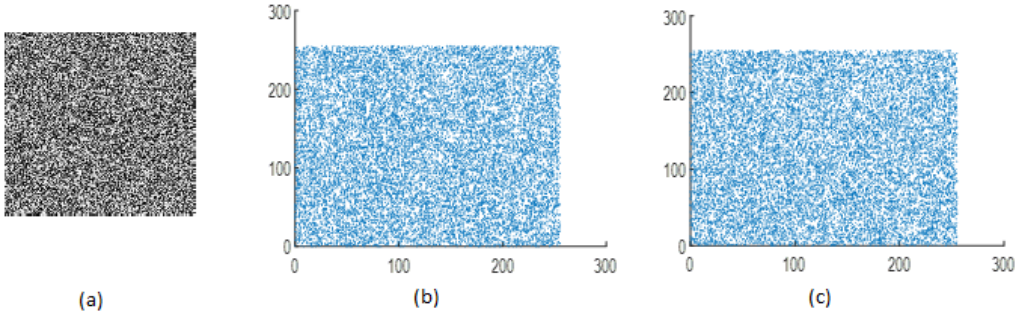


Fig. 4.2 Correlation in encrypted image 'Lena' in different directions (a) Encrypted Lena (b) Horizontal correlation (c) Vertical correlation

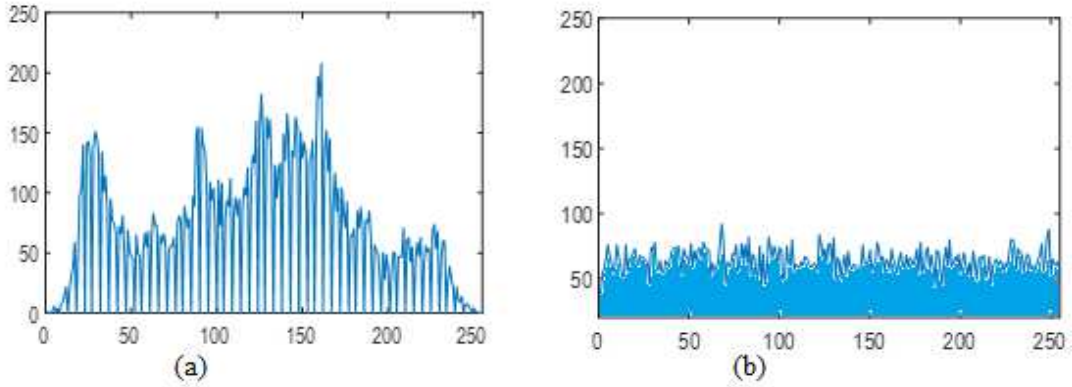


Fig. 4.3 Histograms of 'Lena' image (a) Lena (b) Encrypted Lena

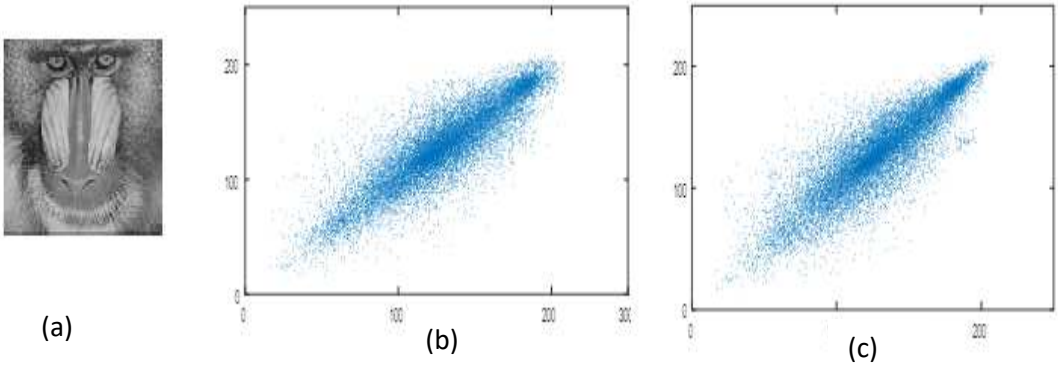
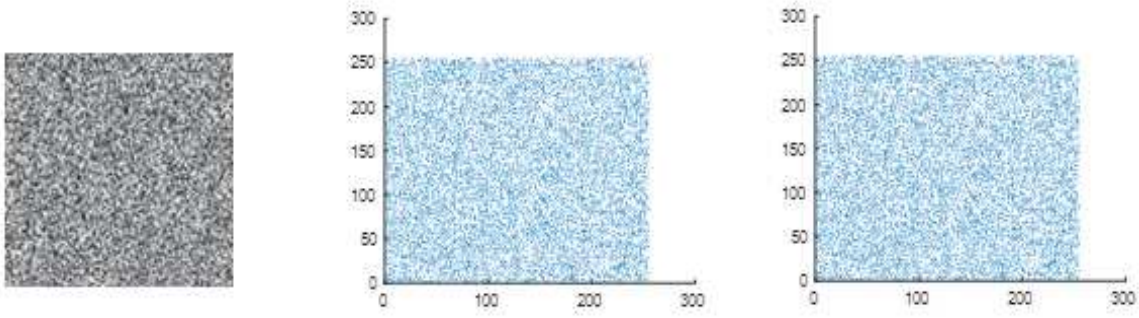
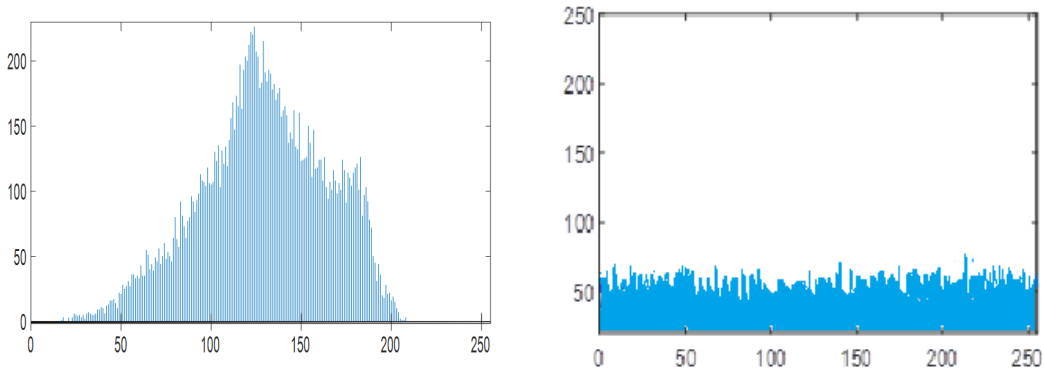


Fig 4.4 Correlation in plaintext image 'Baboon' in different directions (a) Image Baboon (b) Horizontal correlation (c) Vertical correlation



(a) (b) (c)

Fig.4.5 Correlation in encrypted image 'Baboon' in different directions (a) Encrypted Baboon(b) Horizontal correlation (c) Vertical correlation



(a) (b)

Fig. 4.6 Histograms of 'Baboon' image (a) Baboon (b) Encrypted Baboon

Entropy is another statistical measure which characterizes the information content of an image. It is a scalar quantity and is measured for a gray image as

$$H = -\sum_{k=1}^{K} P_k \log_2 P_k \qquad (4.1)$$

where K is the number of gray levels and P_k denotes probability associated with the gray level k .

The calculation using equation (4.1) shows that the entropy of the cover image Lena is 7.3883 whereas, for its encrypted version, it is 7.9599. For cover image Baboon, these values are 7.1297 and 7.9612 respectively. It is observed that for the encrypted version, the entropy value closely approaches its ideal value 8 implying negligible leakage of texture information. The above analysis proves that the Paillier scheme implemented here can effectively eliminate the correlation and makes the encrypted data secure against statistical attacks.

4.3 Encrypted spatial domain RDH Algorithm

An Algorithm is proposed in this thesis to perform data embedding in each pixel of the encrypted image and perfect reconstruction of the cover image and secret data. The flowchart given below describes the basic steps of implementation of the algorithm.

Detailed steps for embedding and extraction process of RDH algorithm presented in this thesis are given below:

The cover image is pre-processed before encryption in this algorithm to avoid the overflow problem.

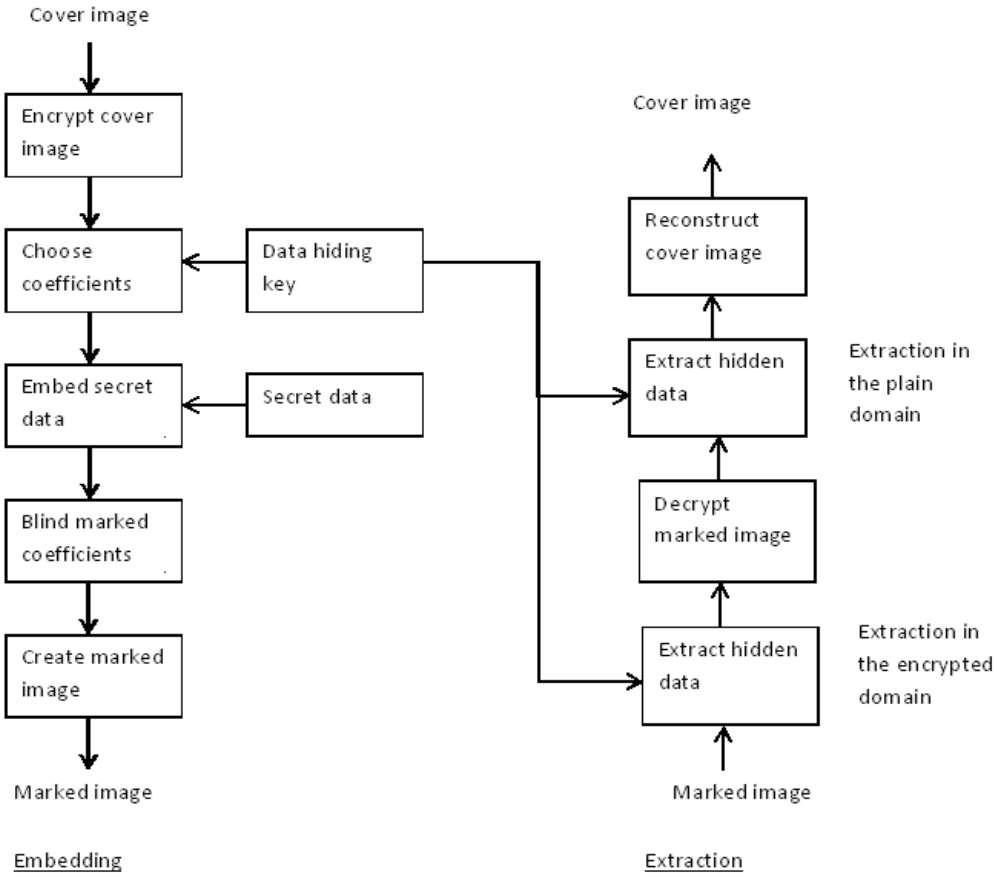


Fig.4.7 Embedding and extraction in the encrypted spatial domain

Embedding:

1. Encrypt the cover image $\mathbf{X}$ using Paillier encryption scheme with equation (2.1) to get $\mathbf{A} = E[\mathbf{X}]$
2. Select encrypted pixel values $\mathbf{A}(i, j)$ using the watermarking key to embed watermark bits.
3. Modify $\mathbf{A}(i, j)$ as $\mathbf{A}_w(i, j)$ to carry the watermark bit b ($b \in \{0,1\}$) as given below

$$\mathbf{A}_w(i,j)=\begin{cases}[\mathbf{A}(i,j)]^2\cdot E[1]\bmod N^2 & \text{if } b=1\\ [\mathbf{A}(i,j)]^2\bmod N^2 & \text{if } b=0\end{cases} \tag{4.2}$$

where $E[1]$ is the encrypted value of integer 1 generated using the same public key.

4. Blind $\mathbf{A}_w(i,j)$ with various values of blinding variable r until the following equation is satisfied

$$\mathbf{A}_w(i,j)=\mathbf{A}_w(i,j)\cdot r^N \bmod N^2 \quad \begin{Bmatrix}\text{is odd if b}=1\\ \text{is even if b}=0\end{Bmatrix} \tag{4.3}$$

Extraction in the encrypted domain:

1. Identify embedding positions using watermarking key
2. From the embedding positions reconstruct the watermark bits as

$$b=\mathbf{A}_w(i,j)\bmod 2 \tag{4.4}$$

3. Perform Paillier decryption using (2.2) to get the modified pixel values in plaintext domain $\mathbf{X}'_w=D[\mathbf{A}_w]$
4. Reconstruct the marked pixel values $\mathbf{X}_w$ as

$$\mathbf{X}_w(i,j)=\left\lceil\frac{\mathbf{X}'_w(i,j)}{2}\right\rceil \tag{4.5}$$

5. Reconstruct original cover image pixel values as

$$\mathbf{X}(i,j)=\left\lfloor\frac{\mathbf{X}'_w(i,j)}{2}\right\rfloor \tag{4.6}$$

Extraction in the plain domain:

1. Perform decryption using equation (2.2) to get the modified image pixels in the plain domain

$$\mathbf{X}'_w=D[\mathbf{A}_w] \tag{4.7}$$

2. Using the watermarking key, recover the watermark bits from marked positions as

$$b = \mathbf{X}'_w(i, j) \bmod 2 \quad (4.8)$$

3. Reconstruct the marked image pixels $\mathbf{X}_w$ using equation (4.5)
4. Recover the original cover pixel values $\mathbf{X}$ using equation (4.6)

symbol $\lfloor . \rfloor$ represents floor operation and $\lceil . \rceil$ represents ceil operation in the above equations.

4.4 Implementation of Algorithm and Performance analysis

Popular gray images Lena and Baboon of size 128x128 are used as the cover in the simulation. Secret data to be embedded is a sequence of binary bits which represent the owner's private information. All the algorithms presented in this thesis have been implemented in C++ with the help of the GNU Multi-Precision library and the NTL library for processing integers of arbitrary length.

Image fidelity and embedding capacity are considered as the key parameters to assess the performance of a watermarking algorithm. Peak Signal to Noise Ratio (PSNR) is a commonly used quality metric to evaluate the quality of the watermarked image with respect to the cover image. If $\mathbf{X}$ represents the original cover image and $\mathbf{X}_w$ represents the watermarked image then PSNR is defined in terms of Mean Squared Error (MSE) as given below.

$$PSNR(dB) = 10\log_{10}\frac{255^2}{MSE} \quad (4.9)$$

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{J=1}^{N}\left[\mathbf{X}(i, j) - \mathbf{X}_w(i, j)\right]^2 \quad (4.10)$$

where M and N represent the dimension of the cover image. A high value of PSNR denotes the efficiency of the watermarking algorithm.

Embedding capacity (or payload) of a watermarking algorithm is usually measured in bits per pixel (bpp). An embedding capacity of 1 bpp indicates that every pixel of a cover image carries one watermark bit. It is well understood that each watermark bit inserted into an image will generally distort the image, or in other words, PSNR is having an inverse relationship with embedding capacity. One needs to make a compromise on one parameter to improve the other.

The proposed algorithm is simple and needs only elementary arithmetic operations. There is no further data expansion due to watermarking other than the expansion caused by the encryption, which is unavoidable for preserving privacy. The proposed Algorithm permits data embedding in each pixel of the encrypted image and allows perfect reconstruction of the cover image and secret data. The preprocessing is a simple task and its output is very similar to the input with a PSNR of 90.28 dB. Even with a high embedding rate of 1 bpp, the algorithm provides high perceptual quality for the marked image with a PSNR of 51.14 dB which is still better in comparison with recently reported algorithms that offer a lesser embedding capacity. Fig. 4.8 shows the visual quality of marked images for different embedding rate.

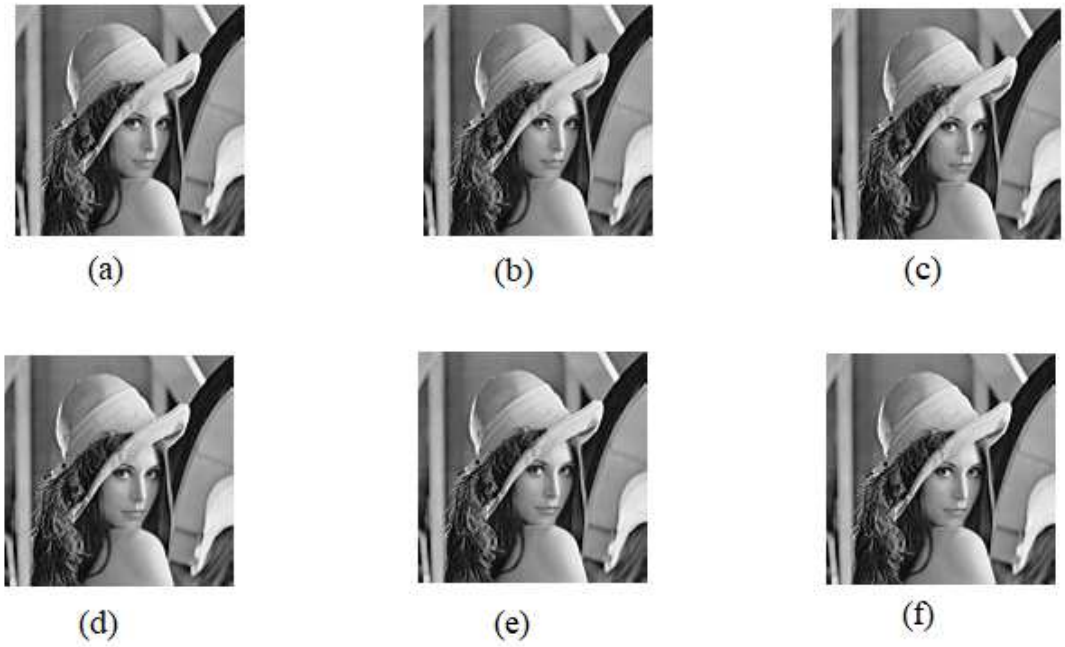


Fig. 4.8 Cover and watermarked images with Algorithm 1 for different embedding rates: (a) cover image (b) 1/16 bpp (c) 1/8 bpp (d) 1/4 bpp (e) 1/2 bpp (f) 1 bpp

The Fig.4.9 shows the variation of PSNR for different embedding rate with the proposed algorithm and is compared with existing algorithms that use different symmetric encryption schemes on the cover image Lena. It is obvious that the average embedding rate in these algorithms except [59], is less than 1 bpp. The CPE-HCRDH scheme in [59] offers 1bpp embedding rate but introduces distortion during the reconstruction of the cover image. It is quite evident from the figure that the proposed algorithm has a better performance compared to these schemes both in terms of embedding rate and PSNR of the marked images.

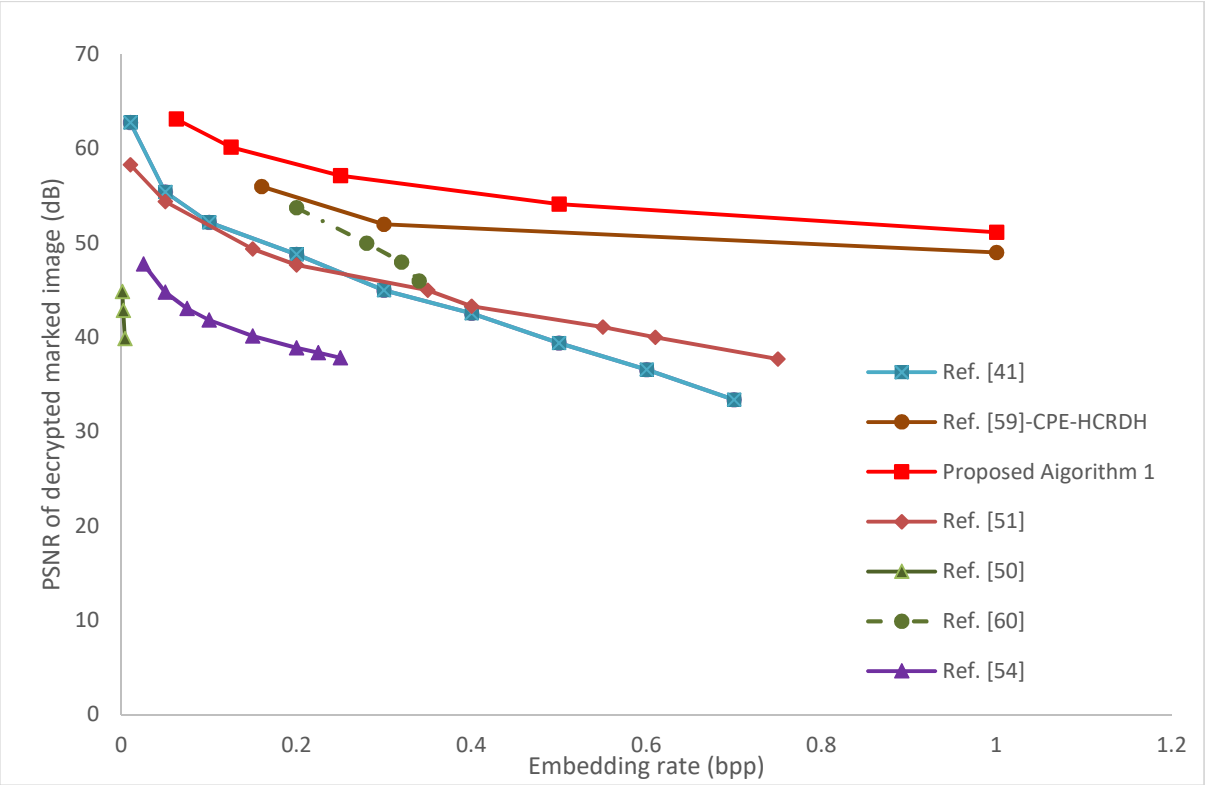


Fig. 4.9 Embedding rate-PSNR performance comparison of proposed Algorithm 1on cover image Lena with symmetric encryption based methods

Performance comparison of the algorithm with lossless schemes that use Paillier encryption [66-69] is given in Fig.4.10.It is clear that the proposed algorithm supports a higher embedding rate along with improvement in the quality of the marked image.

To check the consistency of the performance of the algorithm, it is implemented on a different cover image Baboon and the results are compared with some of the state-of-the-art algorithms that used the same cover image. The results are given in Fig. 4.11 and 4.12. As expected, the proposed algorithm gives similar results on a different cover image which is better than the performance of many of the state-of-the-art algorithms.

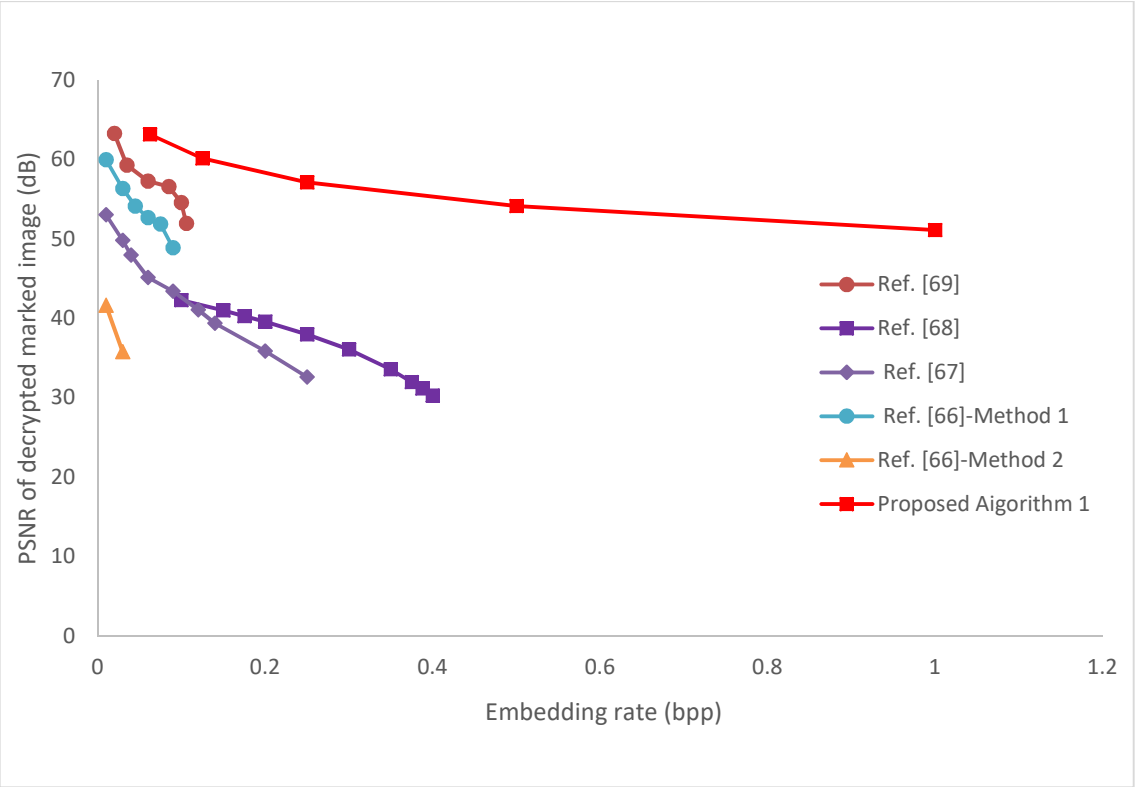


Fig.4.10 Embedding rate-PSNR performance comparison of Algorithm 1 with Paillier encrypted schemes on image Lena

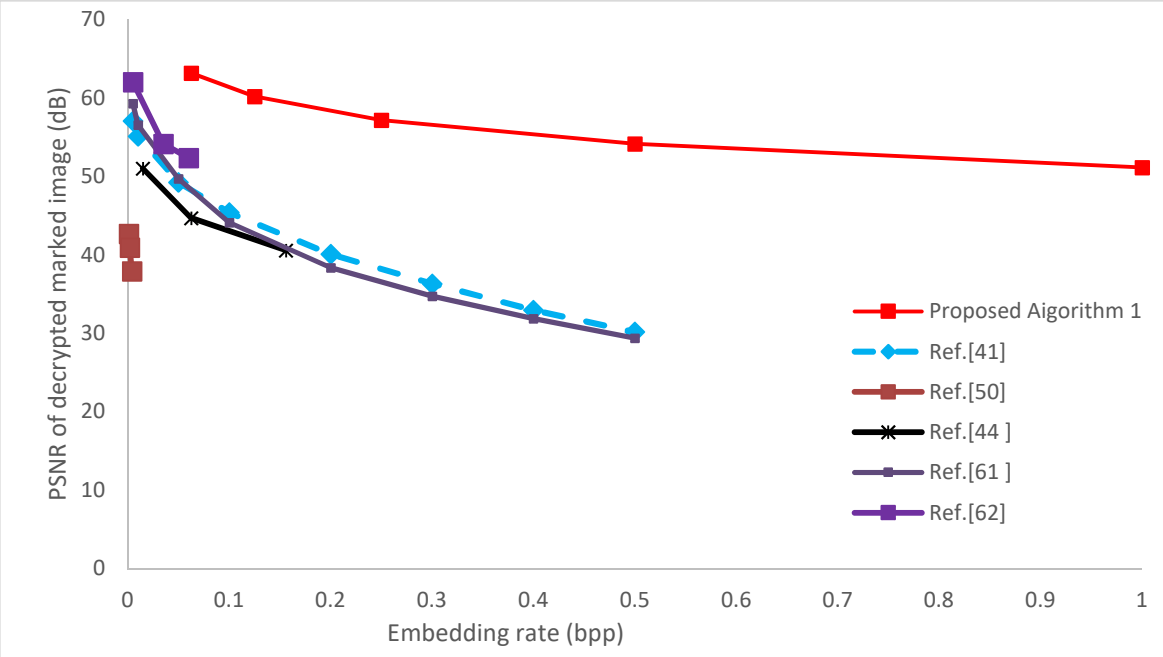


Fig.4.11 Embedding rate-PSNR performance comparison of Algorithm 1 with symmetric encryption schemes on image Baboon.

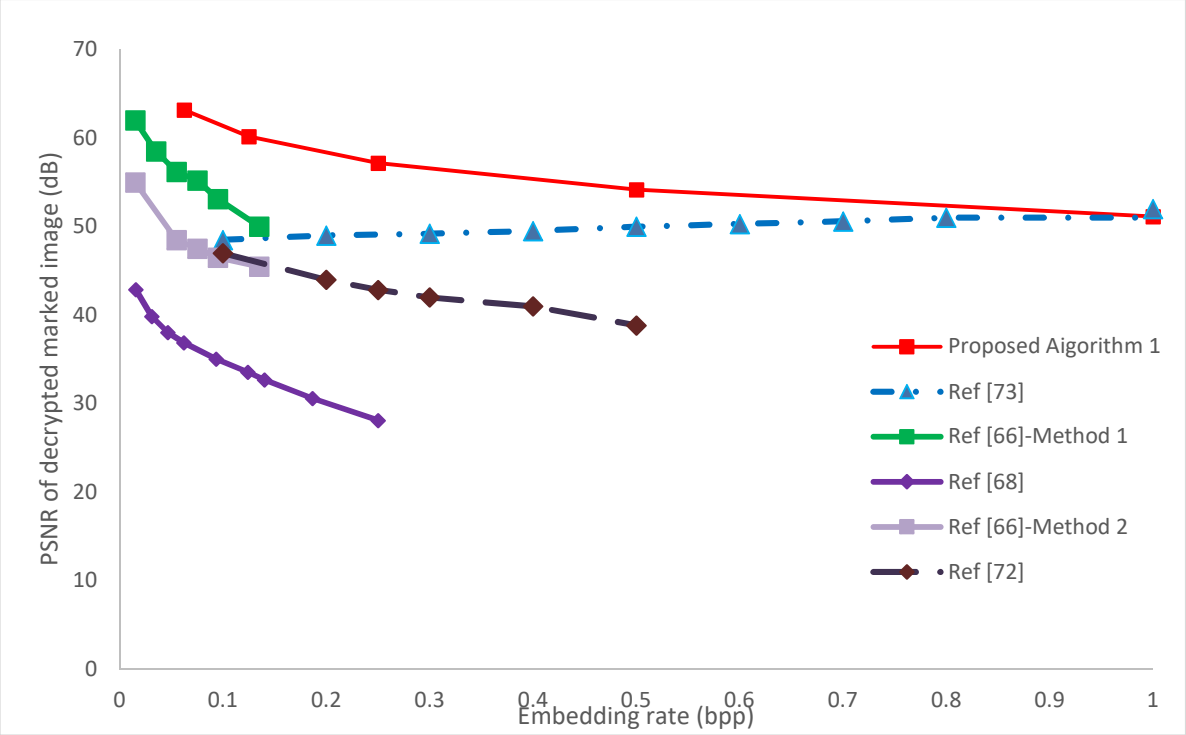


Fig.4.12 Embedding rate-PSNR performance comparison of Algorithm 1 with Paillier encrypted schemes on image Baboon.

To further evaluate the quality of embedded images, the popular Structural Similarity Index Metric (SSIM) is utilized. SSIM between two images W and X is computed as:

$$\text{SSIM}(\text{W},\text{X}) == \frac{(2\,\mu_w\mu_x+C_1)(2\sigma_{WX}+C_2)}{(\mu_W^2+\mu_X^2+C_1)(\sigma_W^2+\sigma_X^2+C_2\,)} \tag{4.11}$$

where $\mu_W, \sigma_W^2, \mu_X, \sigma_X^2$ denote the mean and standard deviation of W and X respectively, σ_{WX} represents the covariance between W and X and C_1 and C_2 are small constants near zero.

The range of SSIM values lies between -1 and 1.When the two input data sets are exactly the same, the SSIM output is 1. Table 4.2 provides the SSIM values of watermarked images for different embedding rates on different cover images.

Table 4.2 SSIM values of the proposed scheme for different embedding rates.

Cover image	Embedding rate(bpp)	SSIM
Lena	0.0625	0.9998
	0.125	0.9997
	0.25	0.9996
	0.5	0.9994
Baboon	0.0625	0.9999
	0.125	0.9998
	0.25	0.9998
	0.5	0.9998

4.5 Summary

This chapter provides the details of the proposed algorithm implemented in the encrypted domain. The algorithm is developed based on the findings of Guo *et* al.[84] where the authors prove that process of embedding does not depend on the pictorial information in the cover. The analysis proves that the performance of the proposed algorithm does not depend on the pictorial information of the cover image used, thus providing an embedding capacity which is independent of the cover image. The encryption is strong enough and does not reveal anything about the content of the cover image and thus protects privacy. The PSNR and SSIM values obtained for the marked image for various embedding capacity show that the marked images show a close resemblance to the original cover. Thus, unlike many of the existing algorithms, the proposed algorithm provides full embedding capacity with high perceptual quality. The next chapter explains the details of proposed algorithms in the transform domain.

CHAPTER 5
RDH IN THE ENCRYPTED TRANSFORM DOMAIN

Data hiding techniques implemented in the spatial domain are more susceptible to attacks compared to transform domain implementations. RDH when implemented in the encrypted transform domain, inherit all the advantages of their spatial domain counterpart.

The chapter explains the proposed reversible data hiding algorithms in the encrypted transform domain. It also gives a brief description of the implementation aspects of DCT in the encrypted domain. The performances of the algorithms are compared with the similar implementations available in the literature.

5.1 Introduction

Spatial domain data hiding techniques are easy to implement and generally require a lower computational cost. But compared to the data hiding methods in the transform domain they are found less robust against tampering. Watermarking schemes implemented in the transform domain are now more popular, as they are more resistant to several attacks and common signal processing operations which may lead to distortions. Transform domain methods hide messages such that they spread on the entire cover image which makes them more robust towards attacks and remain imperceptible to the human sensory system.

In this chapter three algorithms are presented in the encrypted transform domain for data hiding/watermarking applications. All these algorithms are implemented using the encrypted discrete cosine transform (e-DCT). A brief description of the

implementation of DCT in the encrypted domain using the Paillier encryption is given in the following section.

5.2 Discrete cosine transform in the encrypted domain

DCT is a powerful tool in image processing applications. When the input image is provided in the encrypted form in privacy-preserving applications, DCT of the encrypted pixel values can be calculated using the homomorphic properties of the cryptosystem. In [78] authors describe the computation of two dimensional DCT in the encrypted domain. They assume that the chosen cryptosystem is homomorphic with respect to addition and also probabilistic in nature. An example of such a cryptosystem is the Paillier scheme which satisfies both these properties. Further, computing DCT in the encrypted domain requires representing the values of the pixels, the DCT coefficients, and the transformed values as integers on a finite field/ring.

Let $x(n)$ be a signal such that $x(n) \in R, n = 0,1,.....M-1$, and $x(n)$ has been scaled such that $|x(n)| \leq 1$. For computing $x(n)$ in the encrypted domain, the signal values have to be expressed as integers in Z_N. Let $s(n)$ be the integer version of $x(n)$ computed as $s(n) = \lceil Q_1 x(n) \rfloor$, where the symbol $\lceil \ . \rfloor$ indicates rounding operation, and Q_1 is a suitable scaling factor. If the difference between extreme values of $s(n)$ is below N, then it can be reversibly represented in Z_N. If we assume $|s(n)| < \frac{N}{2}$, then $x(n)$ can be estimated from the encrypted $s(n)$ (i.e. $E[s(n)]$) as:

$$\hat{x}(n) = \begin{Bmatrix} \frac{D[E[s(n)]]}{Q_1}, & if\ D[E[s(n)]] < \frac{N}{2} \\ \frac{D[E[s(n)]]-N}{Q_1}, & if\ D[E[s(n)]] > \frac{N}{2} \end{Bmatrix} \quad (5.1)$$

The scaled type II DCT of $x(n)$ may be defined as:

$$X(k) = \sum_{n=0}^{M-1} x(n) cos \frac{\pi(2n+1)k}{2M}, \quad k = 0,1 \ldots., M-1 \tag{5.2}$$

For equation (5.2), integer DCT may be defined as

$$S(k) = \sum_{n=0}^{M-1} C_M(n,k)\, s(n), k = 0,1 \ldots, M-1 \tag{5.3}$$

where$C_M(n,k) = \lceil Q_2 \cos \pi(2n+1)k/2M \rfloor$and $Q_2$is a scaling factor.

Using a similar approach, inverse integer DCT can be defined as:

$$x(n) = \sum_{k=0}^{M-1} c(k) X(k) cos \frac{\pi(2n+1)k}{2M}, \quad n = 0,1 \ldots., M-1 \tag{5.4}$$

where$c(k) = \begin{cases} \frac{1}{2} if\ k = 0 \\ 1\ if\ k \neq 0 \end{cases}$

Since all the computations involve integer values and there is no scaling, the expression (5.4) can be calculated in the encrypted domain using the homomorphic properties of the cryptosystem. For example, if the input is encrypted with the Paillier cryptosystem, e-DCT is computed as

$$E[S(k)] = E[s(n)]^{C_M(n,k)}, k = 0,1.., M-1 \tag{5.5}$$

For processing an image of size $M \times N$, the expression (5.5) can be extended to the two dimensional space.

Thus two-dimensional DCT of input $\mathbf{X}$ of dimension $M \times N$ in plaintext domain is defined as

$$\mathbf{Y}_{k_1 k_2} = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \mathbf{X}_{ij} C_M(i,k_1) C_N(j,k_2) \tag{5.6}$$

where $0 \le k_1 \le M-1$, $0 \le k_2 \le N-1$ and C represents the transform coefficient function.

If $\mathbf{A}$ denotes the encrypted version of $\mathbf{X}$ with Paillier encryption scheme, then two-dimensional DCT can be computed in the encrypted domain as:

$$\mathbf{B}_{k_1 k_2} = \prod_{i=0}^{M-1} \prod_{j=0}^{N-1} \mathbf{A}_{ij}^{C_M(i,k_1) C_N(j,k_2)} \tag{5.7}$$

5.3 Algorithms for RDH in the encrypted transform domain

In this chapter three algorithms with high embedding rate are proposed for embedding the data in the encrypted DCT domain. All these algorithms use the Paillier scheme for image encryption. Additive homomorphic property and probabilistic property of the Paillier scheme are used to embed the secret data in the encrypted cover image. The Algorithm1 allows hidden data extraction only in the plaintext domain whereas Algorithm2 supports the same in the encrypted domain. Algorithm 3 combines the properties of Algorithm1 and 2 and permits secret data extraction both in the plaintext domain and ciphertext domain. These algorithms outperform the existing algorithms in terms of the embedding capacity and provide higher PSNR for the same embedding rate. The hidden data, as well as the original cover, can be extracted without any loss.

5.3.1 Algorithm 1

Algorithm 1 hides the data in the cover by modifying DCT coefficients in the encrypted domain. The retrieval of the data is possible only in the plain domain with a valid data hiding key. Thus only the intended receiver can extract the marked image, secret data, and original cover by applying the private key and data hiding key.

The Fig. 5.1 shows the flowchart that explains the basic embedding and extraction processes.

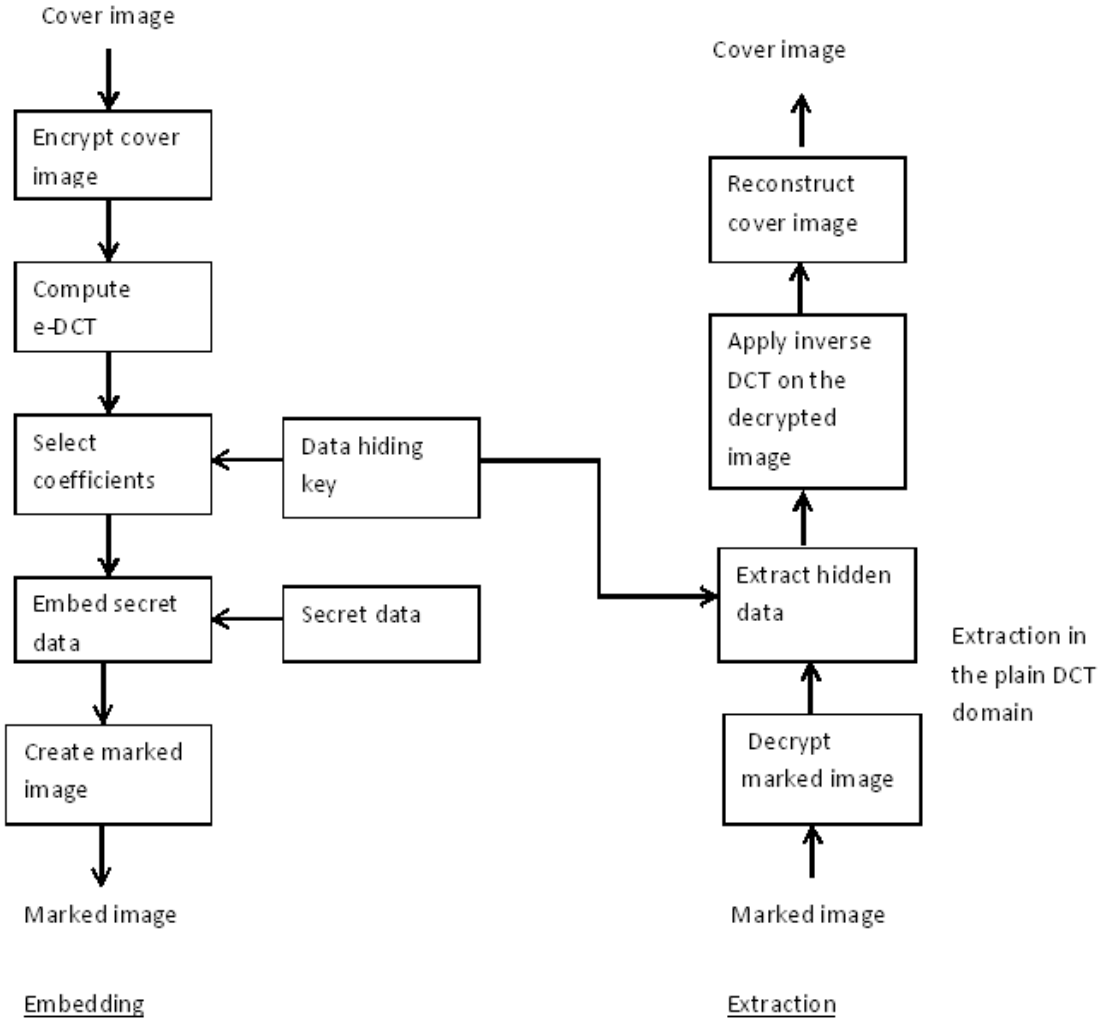


Fig. 5.1 Algorithm1-Embedding and extraction processes

Embedding:

1. Encrypt the cover image $\mathbf{X}$ with the Paillier scheme and a key size of 1024 bits using equation (2.1).
2. Divide the encrypted image into blocks of size 8 x 8 and apply two-dimensional e-DCT using equation (5.7) to obtain $\mathbf{F}$.

3. Using the data hiding key, select the coefficients for modification in $\mathbf{F}$.
4. Perform data hiding by modifying the selected coefficients of $\mathbf{F}$ using the following equation.

$$\mathbf{F}_w(i,j) = \begin{cases} [\mathbf{F}(i,j)]^2 \cdot E[1] \bmod N^2 & \text{if } b = 1 \\ [\mathbf{F}(i,j)]^2 \bmod N^2 & \text{if } b = 0 \end{cases} \quad (5.8)$$

where $E[1]$ is the ciphertext for binary 1. $\mathbf{F}(i,j)$ and $\mathbf{F}_w(i,j)$ represent the original and modified e-DCT coefficients of the cover image and b is the secret bit for hiding.

In plain domain, this operation is equivalent to:

$\mathbf{f}_w(i,j) = 2\mathbf{f}(i,j) + b$ which generates either an odd value or even value of DCT coefficients for b=1 and b=0 respectively. $\mathbf{f}(i,j)$ and $\mathbf{f}_w(i.j)$ represent original and modified DCT coefficients in plain domain.

Extraction:

1. Perform Paillier decryption of modified e-DCT coefficients to compute DCT coefficients in the plain domain ($\mathbf{f}_w(i,j)$)
2. Extract hidden data using data hiding key as

$$b = \mathbf{f}_w(i,j) \bmod 2 \quad (5.9)$$

3. To reconstruct the cover, original DCT values are computed through equation

$$\mathbf{f}(i,j) = \left\lfloor \frac{\mathbf{f}_w(i,j)}{2} \right\rfloor \quad (5.10)$$

where symbol $\lfloor . \rfloor$ represents floor operation.

4. Apply inverse DCT in plaintext domain on $\mathbf{f}$ to reveal the original gray values of the cover medium

5. Perform inverse DCT in plaintext domain on $\mathbf{f}_w$ to reveal the cover image with embedded hidden data.

5.3.2 Algorithm 2

Algorithm 2 exploits the self-blinding property of the Paillier cryptosystem to hide the data in the encrypted transform domain coefficients. This property helps to change one ciphertext to another without actually changing its decryption value. Here, it is used for suitably tailoring odd/even nature of e-DCT coefficients to hold the secret data.

The following equation explains the property of self-blinding.

If $E[m]$ represents a cipher in the Paillier system, then

$$E[m]\cdot(r^N)\bmod N^2 = E[m]\cdot r^N \bmod N^2 \tag{5.11}$$

and $$D[E[m]\cdot r^N] = m \bmod N \tag{5.12}$$

The flowchart given in Fig. 5.2 explains the order of steps for embedding and extraction processes.

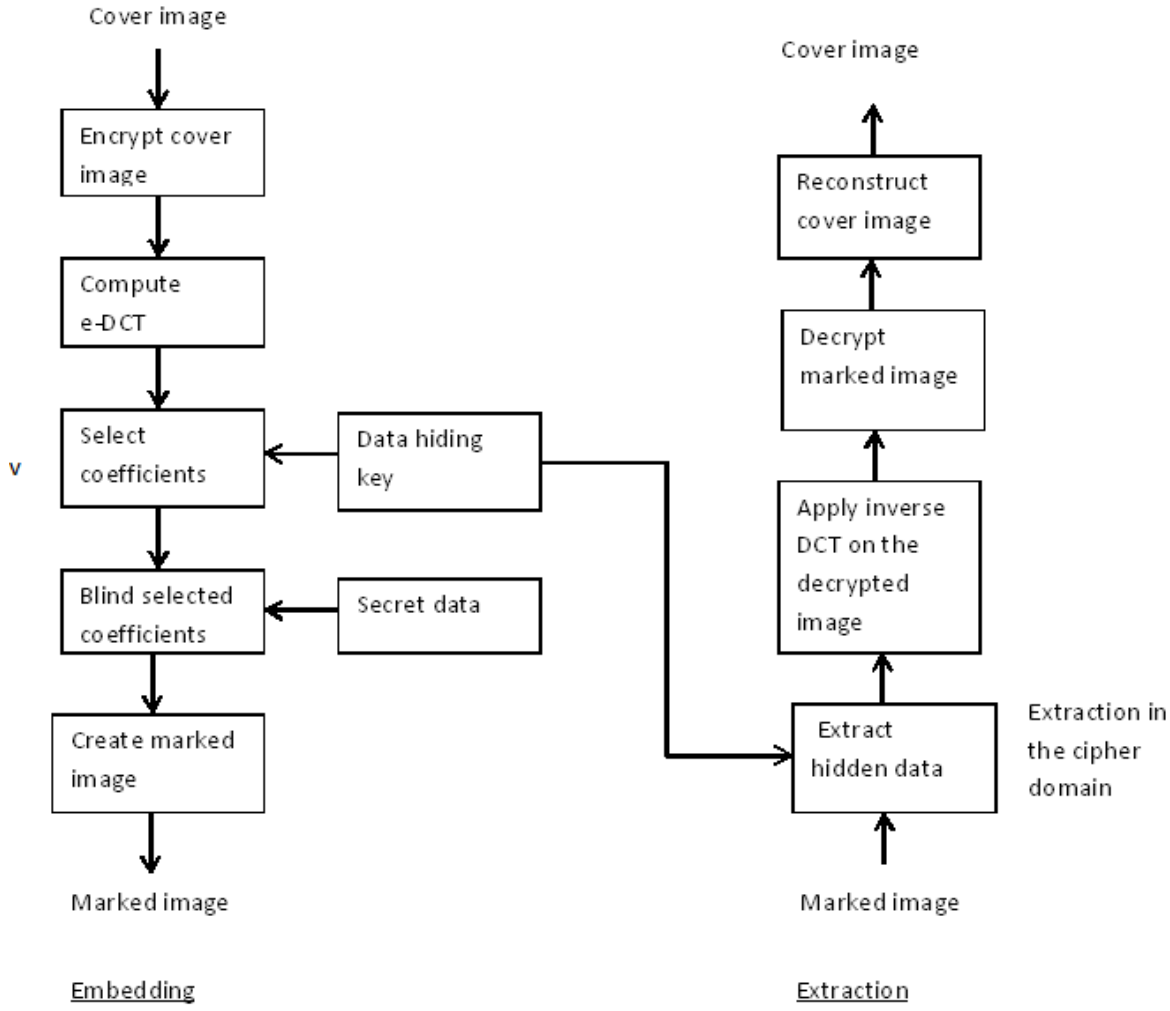


Fig. 5.2 Algorithm 2-Embedding and extraction processes

Embedding:

1. Encrypt the cover image $\mathbf{X}$ using the Paillier encryption with a key size of 1024 bits using equation (2.1) to get $\mathbf{A}$.
2. Divide the encrypted image $\mathbf{A}$ into blocks of size 8x8 and apply two-dimensional e-DCT using equation (5.7) to obtain $\mathbf{F}$.
3. Select the coefficients to be modified in $\mathbf{F}$ using the data hiding key as $\mathbf{F}(i, j)$.

4. To embed a binary bit b into a chosen coefficient, perform self-blinding on $\mathbf{F}(i, j)$ as:

   $\mathbf{F}(i,j) = \mathbf{F}(i,j) \cdot r^N \bmod N^2$ for different values of r until it satisfies the condition

$$b = \mathbf{F}(i, j) \bmod 2 \tag{5.13}$$

Extraction:

1. Identify the modified e-DCT coefficients in $\mathbf{F}$ using the data hiding key.
2. Extract secret data bit b from e-DCT coefficients using equation (5.13).
3. Perform inverse e-DCT on $\mathbf{F}$ followed by Paillier decryption to reconstruct the cover image $\mathbf{X}$.

5.3.3 Algorithm 3

Algorithm 3 embeds watermark bits in the e-DCT domain and permits extraction both in the plaintext domain and cipher domain. The basic steps in the embedding and extraction are shown in the flowchart given in Fig. 5.3.

The detailed steps of embedding and extraction processes are given below.

Embedding:

1. Encrypt the cover image $\mathbf{X}$ using the Paillier encryption scheme with equation (2.1) to get $\mathbf{A}$.
2. Divide the encrypted image $\mathbf{A}$ into blocks of size 8×8 and apply equation (5.7) to compute e-DCT coefficients. Let $\mathbf{B}$ represents a single e-DCT block of size 8×8.
3. Select e-DCT coefficients in each block $\mathbf{B}(i, j)$ using the watermarking key that provides information regarding the blocks and elements in each blocks to be modified.

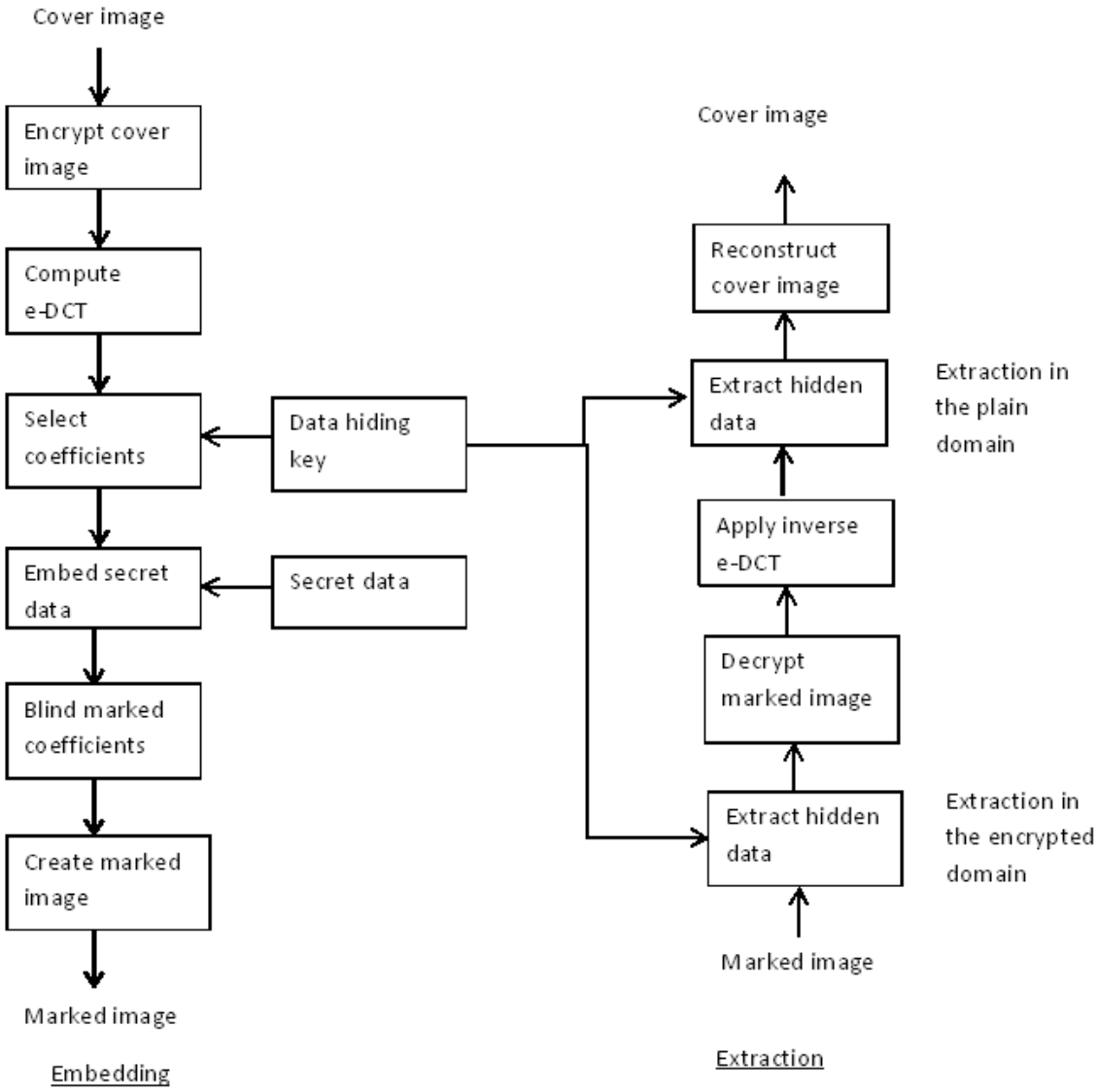


Fig. 5.3 Algorithm 3-Embedding and extraction processes

4. Modify the selected e-DCT coefficients in each block $\mathbf{B}(i, j)$ to $\mathbf{B}_w(i, j)$ using the watermarking key, to carry watermark bit b as follows:

$$\mathbf{B}_w(i, j) = \begin{cases} [\mathbf{B}(i, j)]^2 \cdot E[1] \bmod N^2 & \text{if } b = 1 \\ [\mathbf{B}(\mathrm{i}, \mathrm{j})]^2 \bmod N^2 & \text{if } b = 0 \end{cases} \tag{5.14}$$

Where $E[1]$ denotes the encrypted value of integer 1. The number of blocks selected depends on the size of the watermark.

5. Blind $\mathbf{B}_w(i,j)$ further with various values of blinding variable r until the following equation is satisfied.

$$\mathbf{B}_w(i,j) = \mathbf{B}_w(i,j) \cdot r^N \bmod N^2 = \begin{cases} \text{is odd if } b = 1 \\ \text{is even if } b = 0 \end{cases} \tag{5.15}$$

6. Compute matrix $\mathbf{C}$ concatenating modified blocks $\mathbf{B}_w$

Extraction in the encrypted domain:

1. Using the watermarking key recover the secret bits from marked positions in each block as:

$$b = \mathbf{B}_W(i,j) \bmod 2 \tag{5.16}$$

2. Decrypt $\mathbf{C}$ using equation (2.2) to compute the modified DCT coefficients in the plaintext domain $\mathbf{Y}_w$

3. Obtain the watermarked image $\mathbf{X}_w$ in the plaintext domain by computing inverse DCT of $\mathbf{Y}_w$ by considering 8×8 blocks.

4. Recover unmodified DCT coefficients of the original cover as

$$\mathbf{Y} = \left\lfloor \frac{\mathbf{Y}_w}{2} \right\rfloor \tag{5.17}$$

5. Reconstruct original cover $\mathbf{X}$ from $\mathbf{Y}$ through inverse DCT in the plaintext domain by considering 8×8 blocks

Extraction in the plain domain:

1. Decrypt $\mathbf{C}$ using equation (2.2) to compute the modified DCT coefficients in the plaintext domain as $\mathbf{Y}_w$

2. Using the watermarking key recover the hidden bits from marked positions in each block $\mathbf{f}$ of $\mathbf{Y}_w$ as using either equation (4.4) or (5.9)
3. Recover the watermarked image $\mathbf{x}_w$ in the plaintext domain by computing inverse DCT of $\mathbf{Y}_w$ by considering 8×8 blocks
4. Recover unmodified DCT coefficients of original cover $\mathbf{Y}$ using equation (5.17)
5. Reconstruct original cover $\mathbf{X}$ from $\mathbf{Y}$ through inverse DCT in the plaintext domain by considering 8×8 blocks

5.4 Implementation of algorithm and Performance analysis

Gray image of 'Lena' of size 128×128 are used as cover images in the simulation of these algorithms. Watermark is a string of binary data. Data embedding is performed on the Paillier encrypted cover images in the e-DCT domain for greater security. Simulation is done in C++ with the help of the GNU Multi-Precision library and the NTL library.

For data hiding algorithms proposed here, the embedding capacity is either one bpp or very close to it. Homomorphic embedding avoids the disclosure of sensitive cover data to an untrusted embedder. Original cover media can be extracted only by the authorized receiver using his private key, which provides enough robustness. The bit error ratio (BER) is zero for the extraction of the hidden secret data in their respective domains of extraction. The retrieved cover is exactly same as the original one with PSNR value infinity.

Fig. 5.4 compares the embedding capacity of the proposed algorithms with the existing algorithms reported in the encrypted transform domain [81, 82].The algorithms proposed by Zheng *et.*al [81] and Wu *et*. al[82] utilise the correlation existing in the chosen cover image for watermark embedding and extraction. It is already proved that

correlation based watermarking algorithms are vulnerable to statistical attacks. Further use of correlation properties for watermarking imposes an upper limit on the embedding capacity. The embedding capacity cannot be greater than 0.5 bpp in these algorithms as modification of every coefficient to carry secret data demands at least another one to be left unaltered for the data extraction. Another disadvantage of using correlation properties of natural images for data hiding is that the embedding capacity varies with the cover image chosen as the extent of correlation will be different in different images. The proposed algorithms outperform in these areas as they are not correlation based and embedding capacity is independent of the cover image chosen. The Algorithms1 & 3 allow the modification of 63 coefficients in a block of 64 elements giving an embedding capacity greater than 0.98 bpp and Algorithm 2 allows modification of all 64 elements providing an embedding capacity of 1bpp.

Fig.5.5 to Fig.5.7 compares PSNR vs. embedding strength performance of these algorithms in a block of size 8×8. For an embedding rate of 1 bit/block, the proposed Algorithm1 and 3gives an improvement of 6.44 dB and 10.95 dB in PSNR values compared to [81] and [82] respectively. The performances in terms of PSNR for algorithms 1 and 3 are exactly the same since the additional operation of self blinding does not contribute any errors to the data. For Algorithm2, the retrieved PSNR is infinity for any embedding rate chosen.

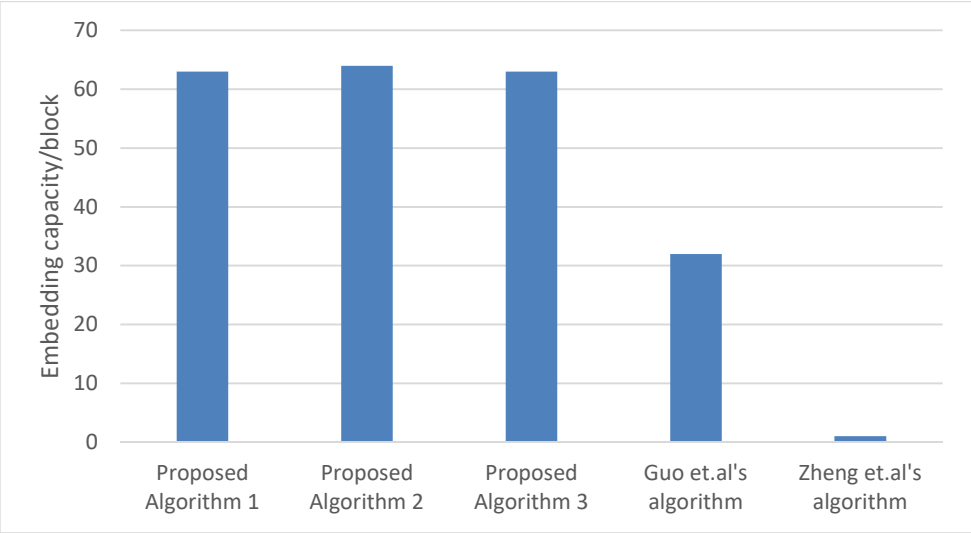


Fig. 5.4 Comparison of embedding capacity/block of transform domain algorithms

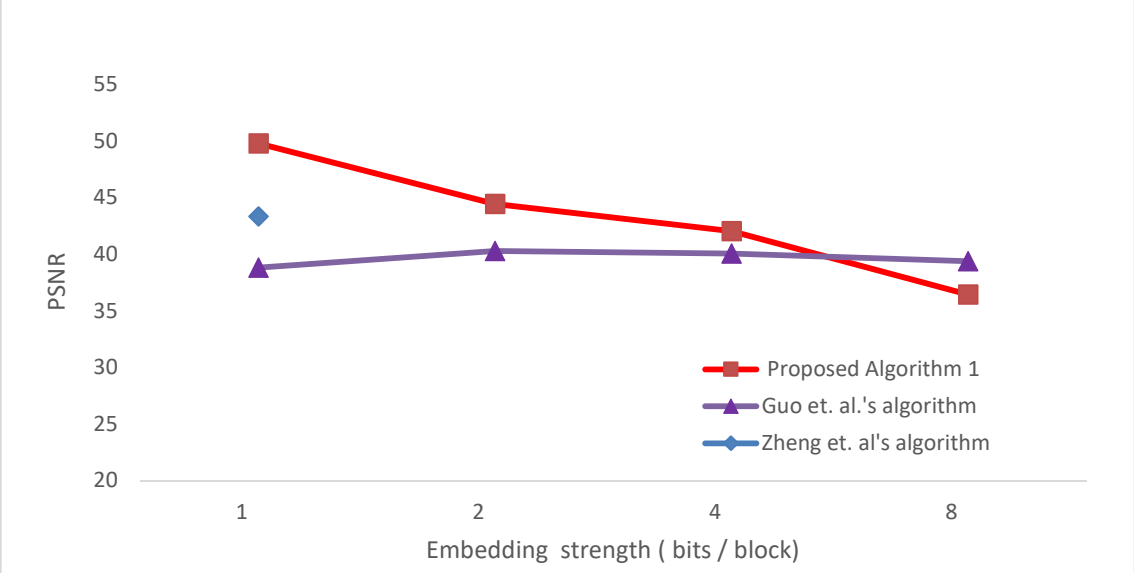


Fig. 5.5 Performance comparison of Algorithm 1 with other transform domain algorithms

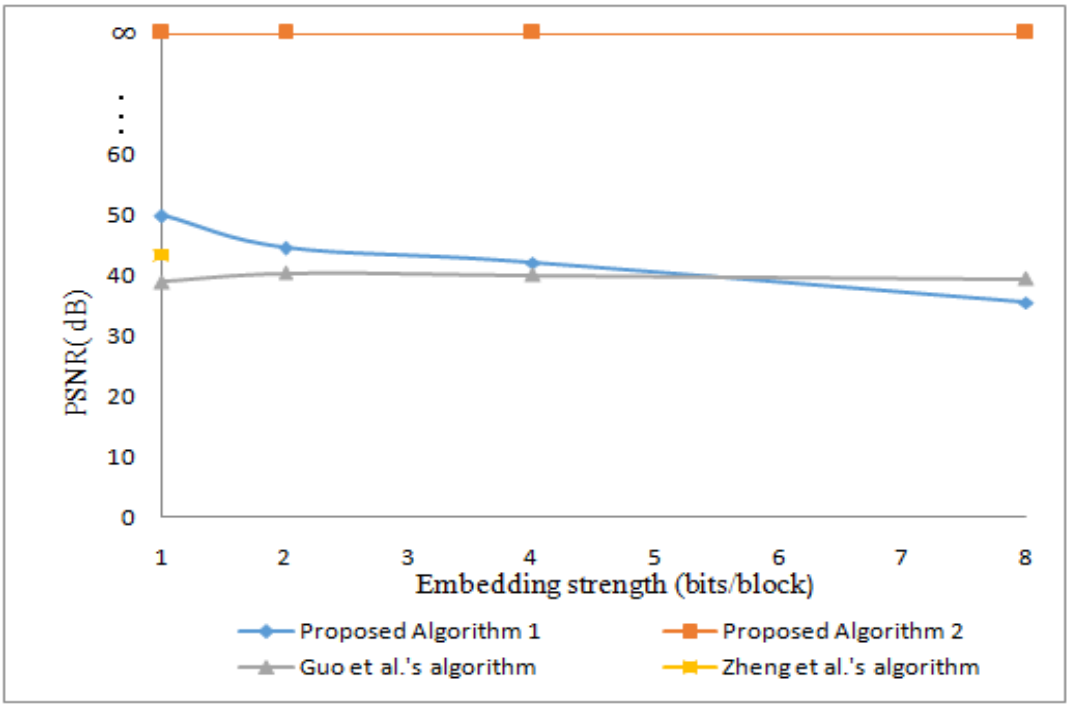


Fig.5.6 Performance comparison of Algorithm1and 2 with other transform domain algorithms

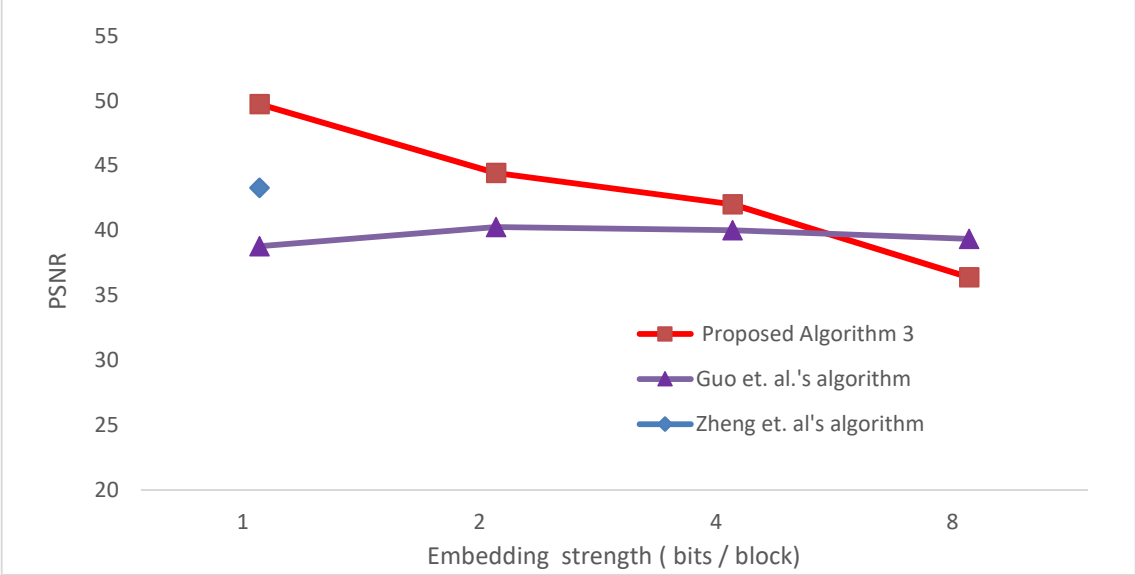


Fig. 5.7 Performance comparison of Algorithm 3 with other transform domain algorithms

The quality of retrieved images can be further evaluated with the help of SSIM measurement. Table5.1 provides the SSIM values of watermarked images for different embedding rates using the proposed schemes.

Table 5.1

SSIM vs. embedding rate of the proposed algorithms

Embedding rate(bpp)	SSIM	
	Algorithm 1 & 3	Algorithm 2
0.0625	0.9975	1
0.125	0.9929	1
0.25	0.9795	1
0.5	0.9501	1

The variation of PSNR with different embedding capacity of the proposed algorithms are given in Table 5.2. As expected, the PSNR values of the retrieved images for different embedding strength remains the same for Algorithm 1 and 3, whereas for Algorithm 2, it is infinity for all embedding strength.

Table 5.2

PSNR vs. Embedding capacity of the proposed algorithms

Embedding capacity (number of bits/block)	PSNR of retrieved images with proposed Algorithm 1 & 3	PSNR of retrieved images with Proposed Algorithm 2
1	49.75	Infinity in all cases
2	44.43	
4	42.03	
8	35.42	
16	30.65	
32	25.66	
48	22.84	
63	19.23	

The perceptual quality of retrieved images using cover image Lena with proposed algorithms are given in Fig. 5.8 to 5.10. The retrieved images give the same quality for Algorithm 1 and 3. We can see that for Algorithm 2, retrieved images exactly resemble the original cover image used.

Fig. 5.8 Retrieved images with Algorithm 1 for different embedding rate: (a) 1/64 bpp (b) 1/32 bpp (c) 1/16 bpp (d) 1/8 bpp (e) 1/4 bpp (f) 1/2 bpp (g) 3/4 bpp (h) 63/64 bpp

Fig. 5.9 Retrieved images with Algorithm 2 for different embedding rate: (a) 1/64 bpp (b) 1/32 bpp (c) 1/16 bpp (d) 1/8 bpp (e) 1/4 bpp (f) 1/2 bpp (g) 3/4 bpp (h) 1 bpp

Fig. 5.10 Retrieved images with Algorithm 3 for different embedding rate: (a) 1/64 bpp (b) 1/32 bpp (c) 1/16 bpp (d) 1/8 bpp (e) 1/4 bpp (f) 1/2 bpp (g) 3/4 bpp (h) 63/64 bpp

5.5 Summary

In this chapter, it is seen that the proposed algorithms clearly outperform the existing transform domain algorithms in terms of embedding capacity and perceptibility of retrieved images. Performance of Algorithm 1 and 3 are similar in terms of PSNR and embedding capacity since self -blinding will not make further distortions to the data. Algorithm1and 3 support an embedding capacity of more than 0.98 bpp whereas Algorithm 2 allows embedding in all the DCT coefficients providing an embedding capacity of 1bpp. The proposed algorithms are completely reversible as they allow perfect

retrieval of the cover image used. The algorithms are simple to implement and do not add further computational complexity. There is no further data expansion other than that is required by the encryption scheme which is inevitable for protecting the privacy of the original cover image details.

The robustness of the proposed algorithms towards various attacks is very significant while evaluating the performance of these algorithms and is discussed in the next chapter.

CHAPTER 6

ATTACKS ON WATERMARKS IN THE ENCRYPTED DOMAIN

Performance of the algorithms towards common attacks in the encrypted domain has been studied in this chapter. Assuming that embedding algorithm is secure and a brute-force attack is practically impossible with the large key size used, the effect of a few attacks that are familiar in plain text domain are implemented in the encrypted domain and their effects on the retrieved images are analysed here.

6.1 Introduction

Encrypted images are not completely free from attacks. A marked image in the encrypted format is still susceptible to certain common signal processing operations and attacks even without the knowledge of the key used for encryption. It is not possible to implement all the plaintext based popular attacks in the encrypted domain. The type of attacks possible in the encrypted domain depends on the nature of the cryptosystem used. Some common attacks in the encrypted domain are reported by Guo et al. in [82]. In this thesis, effect of few such attacks is evaluated, assuming that the potential hacker has no access to the plaintext version of the watermarked data and the watermarking algorithm is secure by itself. These attacks were performed on the marked images in the encrypted spatial domain and encrypted transform domain using the proposed algorithms on cover image Lena. Since the self-blinding procedure does not bring any error to the data, it will not make any further contribution towards the reduction of PSNR in these algorithms.

6.2 Additive noise attack

Being additively homomorphic in nature, encryption with Paillier cryptosystem allows a hacker to easily introduce noise to the watermarked images to destroy the embedded watermark. It can be accomplished as:

$$E[\mathbf{X}'_{i,j}] = E[\mathbf{X}_{i,j}]\,E[\mathbf{n}_k] \mod N^2 \tag{6.1}$$

where $E[\mathbf{n}_k]$ is the noise value and $E[\mathbf{X}'_{i,j}]$ and $E[\mathbf{X}_{i,j}]$ represent tampered and original watermarked pixel values in the encrypted domain. To observe the efficacy of the algorithm towards random noise attack, 1% of noise is added to random locations of the watermarked data using (6.1), with $\mathbf{n}_k$ taking random grey image pixel values. The PSNR of the retrieved watermarked image is measured taking the noiseless watermarked image as the reference. The average PSNR from 10 trials obtained is 30.61dB for encrypted spatial domain implementation and 29.10 dB for encrypted transform domain implementation.

6.3. Salt and Pepper noise

Addition of salt and pepper noise appears as black spots in the white regions and white spots in black regions in a plaintext image. A similar attack is implementable in the encrypted domain using (6.1) where $\mathbf{n}_k$ takes only two possible values, 0 and 255 representing black and white pixel values. A watermarked image with 1% salt and pepper noise gives an average PSNR of 28.55 dB for spatial domain implementation and 25.54 dB for transform domain implementation for 10 trials.

6.4. Cropping attack

Cropping attack intends to strip off some portions of the image to spoil the embedded watermark. When applied in the encrypted domain, the hacker replaces the chosen encrypted pixels with an encrypted zero ($E(0)$). The attack applied here crops the watermarked image along the borders with a band of 8 pixels wide.

6.5. Scaling attack

Image scaling either enlarge or reduce the size of an image. Nearest-neighbour interpolation based scaling attacks can be easily implemented on a Paillier encrypted image. If (M,N) denotes the dimension of the original image, then the size of its scaled version will be (M',N') such that $M' = \alpha M$ and $N' = \alpha N$ where α is the scaling factor. The pixel value of the scaled image I' at coordinates (x', y') is mapped to the pixel value at coordinates (x, y) of the original image I as:

$$E[\, I'_{x',y'}] = E\,[I_{\lfloor \alpha x \rfloor, \lfloor \alpha y \rfloor}] \tag{6.2}$$

The scaling attack implemented here use $\alpha = 0.5$. The symbol $\lfloor . \rfloor$ represents rounding operation.

6.6 Simulation results

The retrieved images corresponding to the aforementioned attacks with spatial domain embedding and transform domain embedding are given in Fig. 6.1 to 6.4. The images that undergo attacks were originally encrypted and embedded with 1024 watermark bits. The decrypted images after these attacks will help the authentic receiver to perceive the type and depth of attacks performed on the encrypted marked image.

Random noise 1%

Salt and pepper 1%

Fig. 6.1 Retrieved images affected by noise attacks in spatial domain embedding

Random noise 1%

Salt and pepper 1%

Fig. 6.2 Retrieved images affected by noise attacks in transform domain embedding

Cropping

Scaling

Fig. 6.3 Retrieved images affected by geometrical attacks in spatial domain embedding

Cropping

Scaling

Fig.6.4 Retrieved images affected by geometrical attacks in transform domain embedding

6.7 Summary

This chapter investigates the possibility of modification of the encrypted marked image by an untrusted person who doesn't have the knowledge of the private key of the intended receiver. Common signal processing based attacks like cutting, cropping, scaling, and noise addition are implemented homomorphically using the encryption key which is not a secret. However, in most of the cases, decrypted images will help the receiver to perceive the type and depth of attacks performed on the encrypted marked image. The cropping attack in the encrypted transform domain leaves a completely distorted image as the effect of cropping will be spread on to the entire image during its reconstruction.

CHAPTER 7
CONCLUSION AND FUTURE SCOPE

7.1 Concluding remarks

The popularity of the internet and the acceptance of social networks like Facebook, WhatsApp, etc. have changed the communication style of common people. A huge amount of personal data has been shared every second through these networks without any concern for their security. The evolution of cloud computing imposes further security challenges to the existing scenario with regard to the privacy and security of the data outsourced. Digital watermarking has been considered as an effective tool to protect Digital Management Rights of digital data that has been communicated through public networks like the internet. The requirement of retrieving the cover media along with the authentication code paved the way towards the evolution of RDH systems. When outsourcing the data as well as services became a common practice in the cloud environment, the security of the data not only during the time of transmission or at storage but during processing also turns to be inevitable.

The need for privacy when outsourcing the data and processes in a cloud environment is a moral and legal right of its customers. In a cloud, privacy cannot be guaranteed without security. The conventional cryptographic primitives such as encryption fail to protect the data during their processing. This raises serious concerns about preserving the privacy of the data when the cover carries very sensitive information that cannot be exposed to an untrusted party even when outsourcing is required.

Secure signal processing techniques have been recently evolved and some privacy preserving techniques have been put forward to solve some of the issues related to secrecy and privacy. These techniques find great applications in a distributed environment like a

cloud where involved parties in a process lack trust in each other. From the available privacy preserving techniques, homomorphic computation is found to be the most promising one to process signals without compromising privacy and security.

Homomorphic cryptosystems allow processing directly on the encrypted data. The nature of operations that can be performed depends on the encryption scheme used. Many of the public key cryptosystems exhibit homomorphism. Currently, additive homomorphism is widely used in privacy preserving implementations in the cloud environment.

In this thesis, four algorithms are presented for secure watermarking applications suitable for the cloud environment. For all the algorithms, the input cover image from the data owner is a gray image encrypted with the Paillier cryptosystem and the embedding is performed either in the encrypted spatial domain or encrypted transform domain. The secret data to be embedded are selected as a sequence of binary bits. Additive homomorphism exhibited by the Paillier scheme is used to embed the data securely without exposing the details of the cover image to the embedder. The first algorithm performs data embedding in the encrypted spatial domain. It modifies the selected encrypted pixel values of the cover image directly to hide the secret data. The remaining three algorithms perform encrypted transform domain embedding to hide the data sequence. Here encrypted DCT coefficients are first computed directly from the encrypted input image. The DCT coefficients in the encrypted form are then modified to hold the secret data. Self-blinding, which is a property of the cryptosystem chosen, is used to tailor the embedded coefficients either odd or even which facilitates the direct retrieval of the secret data bits in the encrypted domain. The authorised clients with a valid data hiding key can directly retrieve the secret data from the marked encrypted image based on the odd/even property of the chosen data still keeping the details of the original cover

as a secret. An obvious advantage of using blinding is that it gives different representations of the encrypted data holding the plaintext value the same. The proposed spatial domain embedding algorithm and one of the transform domain embedding algorithms support data extraction both in the plaintext domain and ciphertext domain which is an added advantage. There is no further data expansion caused by these embedding algorithms other than the expansion created by encryption which is unavoidable to provide security and privacy. The embedding process involves rudimentary mathematical operations that are easy to implement.

The results show that the performances of the proposed algorithms are much better than the state-of-the-art algorithms in terms of embedding capacity and perceptual quality. The encrypted spatial domain implementation provides a high embedding capacity of 1 bpp along with a high PSNR when compared with many of the existing algorithms in the same domain. The transform domain implementations provide an embedding capacity greater than 0.98 bpp while the other reported transform domain algorithms fail to provide a capacity greater than 0.5 bpp. Further, the choice of cover image doesn't limit the embedding capacity of these algorithms. The algorithms offer better robustness towards statistical attacks as they don't exploit any correlation property existing in the cover image. The cover image and secret data can be recovered without any change in all these implementations.

An attempt is also made to evaluate the performance of the algorithms towards some common signal processing operations like the addition of noise, cutting, cropping, etc. on the watermarked images in the encrypted domain. The decrypted images in these attacks will help the authentic receiver to perceive the type and depth of attacks performed on the watermarked image.

7.2 Future scope

The implementation of all algorithms presented in this thesis uses a grayscale image as the cover media. Since the colour images are widely exchanged through the internet and are stored in public clouds, the proposed algorithms can be implemented in colour images to prove authenticity. Algorithms also find applications in anonymous fingerprinting in collaboration with secure commitment schemes that provide solutions to piracy related issues.

Literature shows the implementation of many other popular image-transforms like DFT, FFT, WHT and DWT in the encrypted domain. Thus there is scope for implementing the proposed algorithms using these transforms independently or in combinations in the encrypted domain for privacy preserving image processing applications.

REFERENCES

[1] J. M. Barton, “Method and apparatus for embedding authentication information within digital data”, U S Patent 5 646 997, Jul. 8, 1997.

[2] F.A.P. Petitcolas, R. J. Anderson, M.G. Kuhn, “Information hiding-a survey”, in: Proc. IEEE, vol. 87, no.7, pp. 1062-1078, 1999.

[3] J. Fridrich, D. Soukal, “Matrix embedding for large payloads”, IEEE Trans. Inf. Secur.Forensics, vol.1, no.3, pp.390-394, 2006.

[4] C. Munuera, “Steganography and error-correcting codes”, Signal Process., vol.87, no.6, pp.1528-1533, 2007.

[5] M. U. Celik, G. Sharma, A. M. Tekalp and E. Saber, “Lossless generalized-LSB data embedding”, IEEE Trans. Image Process, vol.14, no.2, pp.253-266, 2005.

[6] C. W. Honsinger, P. W. Jones, M. Rabbani and J. C. Stoffel, “Lossless Recovery of an Original Image Containing Embedded Data”, US Patent 6 278 791, 2001.

[7] J. Mielikainen, “ LSB matching revisited”, IEEE Signal Processing Letters, vol.13, no.5, pp.285-287,2006.

[8] C. H. Yang, C.Y. Weng, S.Wang and H. M. Sun, “Adaptive data hiding in edge areas of images with spatial LSB domain systems”, IEEE Trans. Info. Forensics and Secur., vol. 3, no.3, pp.488-497, 2008.

[9] H. Sakai, M. Kuribayashi and M. Morii, “Adaptive reversible data hiding for JPEG images”, in: Proc. IEEE Int. Symp. Inf. Theory Appl., pp. 1-6, 2008.

[10] C.-C Chang, C.-C Lin, C.-S Tseng and W.-L. Tai, “Reversible hiding in DCT-based compressed images”, Inf. Sci., vol.177, no. 13, pp.2768-2786, 2007.

[11] C.-C. Lin and P.-F. Shiu, “DCT-based reversible data hiding scheme”, J. Softw., vol. 5. No. 2, pp.214-224, 2010.

[12] K. Wang, Z.-M.Lu and Y.-J.Hu, “A high capacity lossless data hiding scheme for JPEG images”, J. Syst. Softw. Vol. 86, no.7, pp.1965-1975, 2013.

[13] A. Nikolaidis, “Reversible data hiding in JPEG images using zero quantized coefficients”, IET Image Process., vol.9, no.7, pp.560-568, 2015.

[14] S. A. Parah, J. A. Sheikh, N. A. Loan and G. M. Bhat, “Robust and blind watermarking technique in DCT domain using inter-block coefficient differencing”, Digital Signal Processing, vol. 53, pp. 11-24, 2016.

[15] F. Huang, X. Qu, H. J. Kim and J. Huang, "Reversible data hiding in JPEG images", IEEE Trans. Circuits Syst. Video Technol., vol. 26,no.9, pp. 1610-1621, 2016.

[16] L. Kamstra and H. J. A. M. Heijmans, "Reversible data embedding into images using wavelet technique and sorting", IEEE Trans. Image Process., vol.14, no.12, pp.2082-2090, 2005.

[17] X. Wang, X. L. Li, B. Yang and Z. M. Guo, " Efficient generalized integer transform for reversible watermarking", IEEE Signal Precess. Lett., vol.17, no.6, pp.567-570, 2010.

[18] M. Fallahpour and M. H. Sedaaghi, "High capacity lossless data hiding based on histogram modification", IEICE Electron. Exp., vol.4, no.7, pp.205-210, 2007

[19] W. L. Tai, C. M. Yeh, and C. C. Chang, "Reversible data hiding based on histogram modification of pixel differences", IEEE Trans. Circuits Syst. Video Technol., vol. 19, no. 6, pp. 906–910, Jun 2009.

[20] X. L. Li, B. Li, B. Yang, and T. Y. Zeng, "General framework to histogram-shifting-based reversible data hiding", IEEE Trans. Image Process., vol. 22, no. 6, pp. 2181–2191, 2013.

[21] G. Coatrieux, W. Pan, N. Cuppens-Boulahia, F. Cuppens, C. Roux, "Reversible watermarking based on invariant image classification and dynamic histogram shifting", IEEE Trans. Inf. Forensics Secur., vol.8, no. 1, pp.111–120, 2013.

[22] X. Li, B. Li, B. Yang and T. Zeng, "General Framework to Histogram-Shifting-Based Reversible Data Hiding", IEEE Trans. Image Process., vol. 22, no. 6, pp.2181-2191, 2013.

[23] X. Li; W. Zhang; X. Gui, B. Yang, "Efficient Reversible Data Hiding Based on Multiple Histogram Modification", IEEE Trans. Inf. Forensics Secur., vol.10, no. 9, pp.2016–2027, 2015

[24] H. Chen, J. Ni, W. Hong, T.-S. Chen, "Reversible data hiding with contrast enhancement using adaptive histogram shifting and pixel value ordering", Signal Process. Image Commun.vol.46, pp. 1-16, 2016.

[25] K. S. Kim, M. J. Lee, H. Y. Lee, and H. K. Lee, "Reversible data hiding exploiting spatial correlation between sub -sampled images", Pattern Recognit., vol. 42, no. 11, pp. 3083–3096, 2009.

[26] K. A. Khan and S. A. Malik, "A high capacity reversible watermarking approach for authenticating images: exploiting down-sampling, histogram processing and block selection", Inf. Sci., vol. 256, pp. 162-183, 2014.

[27] Y. Cheung and H. Wu, “A sequential quantization strategy for data embedding and integrity verification”, IEEE Trans. Circuits Syst. Video Technol., vol.17, no.8, pp. 1007-1016, 2007.

[28] M. J. Saberian, M.A. Akhaee and F. Marvasti, “An invertible quantization based watermarking approach”, in: Proc. IEEE Int. conf. acoustics, Speech and Signal Processing, Las Vegas, USA, pp.1677-1680, 2008.

[29] L.-T. Ko, J.-E. Chen, Y.-s.Shieh, M. Scalia and T.-Y. Sung, “A novel fractional-discrete-cosine-transform-based reversible watermarking for healthcare information management systems”, Math.Problems Eng.2012, pp. 1-17, 2012.

[30] L.-T. Ko, J.-E. Chen, Y.-s.Shieh, H. C. Hsin and T.-Y. Sung, “Nested quantization index modulation for reversible watermarking and its application to for healthcare information management systems”, Comput. Math.Methods Med. 2012, pp. 1-8, 2012.

[31] J. Lee, Y. Chiou and J. Guo, “Reversible data hiding based on histogram modification of SMVQ indices”, IEEE Trans. Inf. Forensics Secur., vol.5, no. 4, pp.638–648, 2010.

[32] J. Tian, "Reversible Data Embedding Using a Difference Expansion”, IEEE Trans. Circuits and Syst. Video Technol, vol. 13, pp. 890-896, 2003.

[33] A. M. Alattar, "Reversible watermark using difference expansion of quads", in: Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing (ICASSP '04). , pp. 377-380, 2004

[34] C.-C. Lee, H.-C.Wu, C.-S.Tsai and Y.-P.Chu, "Adaptive lossless steganographic scheme with centralized difference expansion", Pattern Recognition, vol. 41, pp. 2097-2106, 2008.

[35] F. Peng, Y. Z. Lei, M. Long and X. M. Sun, “A reversible watermarking scheme for two-dimensional CAD engineering graphics based on improved difference expansion”, Comput. Aided Des., vol. 43, no.8, pp. 1018-1024, 2011.

[36] B. Ou, X. Li, Y. Zhao, R. Ni and Y.-Q.Shi.“Pairwise prediction error expansion for efficient reversible data hiding”, IEEE Trans. Image Process.Vol.22, no.12, pp.5010-5021, 2013.

[37] J. Zhou and O. C. Au, “Determining the capacity parameters in PEE- based reversible image watermarking”, IEEE Signal Process. Lett., vol.19, no.5, pp. 287-290, 2012.

[38] L.-C. Dragoi and D. Coltuc, “On local prediction based reversible watermarking”, IEEE Trans. Image Process., vol.24, no.4, pp.1244-1246, 2015.

[39] Q. Pei, X. Wang, Y. Li and H. Li, "Adaptive reversible watermarking with improved embedding capacity", J. Syst. Softw., vol.86, no.11, pp.2841-2848, 2013.

[40] B. Ou, X. Li and J. Wang, "High-fidelity reversible data hiding based on pixel-value-ordering and pairwise prediction error expansion", J. Vis. Commun. Image Represent., vol.39, pp.12-23, 2016.

[41] K. Ma, W. Zhang, X. Zhao, N. Yu and F. Li, "Reversible data hiding in encrypted images by reserving room before encryption", IEEE Trans. on Information Forensics and Security, vol. 8, no. 3, pp. 553-562, Mar. 2013.

[42] Zhenjun Tang, Quanfeng Lu, Huan Lao, Chunqiang Yu and Xianquan Zhang, "Error-free reversible data hiding with high capacity in encrypted image",Optik,vol.157, pp. 750-760, March 2018.

[43] Qin Chuan, He Zhihong, Luo Xiangyang and Dong Jing, "Reversible data hiding in encrypted image with separable capability and high embedding capacity", Information Sciences, vol. 465, pp. 285-304, October 2018.

[44] Xiaotian Wu and Wei Sun,"High-capacity reversible data hiding in encrypted images by prediction error-Signal Processing",vol.104,pp. 387–400, Nov.2014.

[45]] W. Zhang, K. Ma and N. Yu, "Reversibility improved data hiding in encrypted images", Signal Process., vol. 94, no.1, pp.118-127, 2014.

[46] Shuang Yi and Yicong Zhou,"Parametric reversible data hiding in encrypted images using adaptive bit-level data embedding and checkerboard based prediction", Signal Processing, vol. 150, pp. 171-182, September 2018.

[47] Chunqiang Yu, Xianquan Zhang, Zhenjun Tang, Yan Chen and Jingyu Huang, "Reversible Data Hiding with Pixel Prediction and Additive Homomorphism for Encrypted Image", Security and Communication Networks, vol.2018, pp.1-13, Article ID 9103418, 2018.

[48] Y.C.Chen, C.W.Shiu, G.Horng, "Encrypted signal-based reversible data hiding with public key cryptosystem", J. Vis. Commun. Image Represent., vol.25, pp. 1164–1170, 2014.

[49] C.-W.Shiu, Yu-Chi Chen and Wien Hong, "Encrypted image-based reversible data hiding with public key cryptography from difference expansion", Signal Processing: Image Communication, vol.39,DOI: 10.1016/j.image.2015.09.014, 2015.

[50] S. Agrawal and M. Kumar, “Mean value based reversible data hiding in encrypted images”, Optik, vol. 130, pp. 922-934, 2017.

[51]] X. Cao, L. Du, X. Wei, D. Meng and X. Guo, “High capacity reversible data hiding in encrypted images by patch-level sparse representation”, IEEE Trans. Cybern., vol. 46, no.5, pp.1132-1143, 2016.

[52] X. Zhang, “Reversible data hiding in encrypted image”, IEEE Signal Process.Lett., vol. 18, no.4, pp.255- 258, 2011.

[53] W. Hong, T.-S.Chen and H.-Y. Wu, “An improved reversible data hiding in encrypted images using side match”, IEEE Signal Process. Lett., vol.19, no.4, pp.199-202, 2012

[54] X. Liao and C. Shu, “Reversible data hiding in encrypted images based on absolute mean difference of multiple neighboring pixels”, J. Vis. Commun. Image Represent. , vol.28, pp.21-27, 2015.

[55] J. Zhou, W. Sun, L. Dong, X. Liu, O. C. AU and Y. Y. Tang, “Secure reversible image data hiding over encrypted domain via key modulation”, IEEE Trans. Circuits. Syst. Video Technol., vol. 26, no. 3, pp.441-452, 2016.

[56] Z. Yin, B. Luo and W. Hong, “Separable and error-free reversible data hiding in encrypted image with high payload”, Sci. World J., vol. 2014, no. 604876, 2014.

[57] X. Zhang, Z. Qian, G. Feng and Y. Ren, “Efficient reversible data hiding in encrypted images”, J. Vis. Commun. Image Represent. , vol.25, pp.322-328, 2014.

[58] S. Zheng, D. Li, D. Hu, D. Ye, L. Wang and J. Wang, “Lossless data hiding algorithm for encrypted images with high capacity”, Multimedia Tools Appl., vol. 75, no. 21, pp. 13765-13778, 2016.

[59] P. Puteaux and W. Puech, ‘An Efficient MSB Prediction-Based Method for High-Capacity Reversible Data Hiding in Encrypted Images”, IEEE Trans. Inf. Forensics Secur., vol. 13, no. 7, pp.1670-1681, DOI: 10.1109/TIPS.2018.2799381,2018.

[60]D. Xiao, Y. Xiang, H. Zheng and Y. Wang, “Separable reversible data hiding in encrypted image based on pixel value ordering and additive homomorphism”, J. Vis. Commun. Image Represent., 45, pp. 1-10, 2017.

[61]P. Singh and B. Raman, “Reversible data hiding for rightful ownership assertion of images in encrypted domain over cloud”, Int. J. Electron. Commun.(AEU), vol.76, pp. 18-35, 2017.

[62] P. Singh and B. Raman, “Reversible data hiding based on Shamir’s secret sharing for color images over cloud”, Information Sciences ,vol. 422, pp77–97, 2018.

[63] Ming Li , Di Xiao, Yushu Zhang , Hai Nan, “Reversible data hiding in encrypted images using cross division and additive homomorphism”, Signal Processing: Image Communication vol. 39, pp. 234–248, 2015.

[64] Dalel Bouslimi , Gouenou Coatrieux , Michel Cozic , Christian Roux, “Data hiding in encrypted images based on predefined watermark embedding before encryption process”, Signal Processing: Image Communication, vol. 47, pp. 263–270, 2016.

[65] Y.-C. Chen, C.-W.Shiu and G. Hong, “Encrypted signal based reversible data hiding with public key cryptosystem”, J. Vis. Commun. Image Represent.,vol. 25, pp. 1164-1170, 2014.

[66] S. Xiang and X. Luo, “Efficient reversible data hiding in encrypted image with public key cryptosystem”, EURASIP J. Advances in Signal Processing, no.59, DOI: 10.1186//s13634-017-0496-6, 2017.

[67] S. Xiang and X. Luo, “Reversible Data Hiding in Encrypted Domain by Mirroring Ciphertext Group”, IEEE Trans. Circuits and Systems for Video Technology, vol.28, no.11, pp. 3099-3110,DOI 10.1109/TCSVT. 2017.2742023,2018.

[68] X. Zhang, J. Long, Z. Wang and H. Cheng, “Lossless and Reversible Data Hiding in Encrypted Images With Public-Key Cryptography”, IEEE Trans. Circuits and Systems for Video Technology, vol.26, no. 9, pp. 1622-1631, 2016.

[69] M. Li and Y. Li, “Histogram shifting in encrypted images with public key cryptosystem for reversible data hiding”, Signal Process. vol.130, pp. 190-196, DOI:10.1016/j.sigpro.2016.07.002, 2016.

[70] H.-T. Wu, Y.-m.Cheung and J. Huang, “Reversible data hiding in Paillier cryptosystem”, J. Vis. Commun. Image Represent.,vol.40, pp.765-771, 2016.

[71] Wei-Liang Tai and Ya-Fen Chang, “Separable Reversible Data Hiding in Encrypted Signals with Public Key Cryptography”, Symmetry, vol.10, no.1,DOI: 10.3390/sym10010023, 2018.

[72] Yu-Chi Chen, Chih-Wei Shiu , Gwoboa Horng, “Encrypted signal-based reversible data hiding with public key cryptosystem”, J. Vis. Commun. Image R.,vol.25, pp.1164–1170, 2014.

[73] Cuiling Jiang, Yilin Pang, "Encrypted image-based reversible data hiding in Paillier cryptosystem", Multimedia Tools and Applications, vol.79, pp. 693-711, DOI: 10.1007/s11042-019-07874-w, 2019.

[74] Xianyi Chen , Haidong Zhong, Lizhi Xiong, and Zhihua Xia, "Improved Encrypted-Signals-Based Reversible Data Hiding Using Code Division Multiplexing and Value Expansion", Security and Communication Networks, vol. 2018, pp. 1-9, Article ID 1326235, DOI: 10.1155/2018/1326235, 2018.

[75] Xiaotian Wu, Bing Chen and Jian Weng, "Reversible data hiding for encrypted signals by homomorphic encryption and signal energy transfer", J. Vis. Commun., vol. 41, pp.58-64, DOI: 10.1016/j.jvcir.2016.09.005, 2016.

[76] T. Bianchi, A. Piva and M. Barni, "Comparison of different FFT implementations in the encrypted Domain", In: Proc. EURASIP EURASIPCO, 2008.

[77] T. Bianchi, A. Piva and M. Barni, "On the implementation of discrete Fourier transform in the encrypted domain", IEEE Trans. Inform. Forensics and Secur.,vol.4, no.1, pp. 86-97, 2009.

[78] T. Bianchi, A. Piva and M. Barni, "Encrypted domain DCT based on homomorphic cryptosystems", EURASIP J. Inform. Secur., Article ID 716357, DOI:10.1155/2009/716357,2009.

[79] P. Zheng and J. Huang, "Implementation of discrete wavelet transform and multiresolution analysis in the encrypted domain", In: Proceedings of the 19th ACM international conference on Multimedia, pp.413-422, DOI: 10.1145/2072298.2072352, 2011.

[80] P. Zheng, J. Huang, "Discrete wavelet transform and data expansion reduction in homomorphic encrypted domain", IEEE Trans. Image Processing, vol. 22, no. 6, pp.2455-2468, 2013.

[81] P. Zheng and J. Huang, "Walsh-Hadamard transform in the homomorphic encrypted domain and its application in image watermarking", Information Hiding, LNCS, vol.7692, pp.240-254, Springer, 2013.

[82] J. Guo, P. Zheng and J. Huang, "Secure Watermarking scheme against watermark attacks in the encrypted domain", J. Vis. Commun. Image Represent., vol.30, pp.125-135, 2015.

[83] W. Stallings, Cryptography and Network Security-principles and practices, Pearson Education, 2006.

[84] P. Paillier, Public –key cryptosystems based on composite degree residuosity classes, Advances in Cryptology-EUROCRYPT'99, LNCS, vol. 1592, pp. 223-238, 1999.

[85] Luka Arezina, The all-connecting thread: Internet usage statistics or 2019, Nov. 19, 2019.

[86] www.ftc.gov/news-events/July 2019.

RESEARCH PUBLICATIONS

Journals:

1. Jeeva K.A and Sheeba V.S, ‘‘High Capacity Reversible Data Hiding in Encrypted Transform Domain for Privacy Protection", International journal on simulation systems, science and technology (IJSSST), vol. 19, no. 6, Dec. 2018.
2. Jeeva K.A and Sheeba V.S, “Privacy preserving Reversible watermarking through self-blinding”, International journal of Bioinformatics Research and Applications (in press).

Conferences:

1. Jeeva K.A and Sheeba V.S, “High Capacity Reversible Data Hiding in Encrypted Transform Domain for Privacy Protection”, in: International Conference ‘SAPIENCE 18’ organised by Sree Narayana Gurukulam College of Engg. Kerala, Sept. 2018.
2. Jeeva K.A and Sheeba V.S, “Reversible watermarking in the encrypted domain through self-blinding”, in: Springer International conference ICSCT’2019, organised by Anurag group of institutions, Hyderabad, July 2019.